Course Contents

- Selected advanced issues
- Query Processing
- Physical design of databases
- Logical design of relational databases
- The SQL Language for Creating and Querying Relational Databases
- Overview of databases and reasons for using them
What is a database?

- Expert system
- Phone book

What about...

A database is a Queryable Store of Information

Proposed Answer:
Interactions with users and administrator
Software system to manage the database, and its

Nicknames: DBMS, Database System

DATABASE MANAGEMENT SYSTEM
Query Processing

Data Manipulation (insert, modify, delete)

Data Definition (Schema)

SYSTEM

CONCEPTUAL FUNCTIONS OF A DATABASE
Support for Data-Based Applications •

Concurrency Control •

Fault Tolerance •

Organized for Fast Access •

Large Amounts of Persistent Storage •

Semantics of a Data Model •

SYSTEM

CONCRETE BENEFITS OF A DATABASE
select * from work where hours < 10
insert into work values("George", 20)
create table work (name string, hours integer)

Respective Illustrations:

Query Language (Questions) •
Data Language (Knowledge) •
Schema Language (Metadata) •

THREE LANGUAGES IN A DATABASE SYSTEM
Critical Questions for Database Analysis

• What kind of structures in reality do we wish to represent?

• What constructs does the data language need to reflect these reality-structures?
ER constituents.

Analyze the domain to be represented in the DB, into its

- Relationships, e.g., marriages, enrollments
- Attributes, e.g., name, location
- Entities, e.g., students, courses

(The "ER Model")

RELATIONSHIPS

ELEMENTARY ONTOLOGY: ENTITIES AND
Declarative, rooted in classical logic

Predominant model in practice today

query language—SQL

diverse applications. Makes for simple yet rich,

Very simple data language, yet can heuristically represent


RELATIONAL MODEL
A relational database is a collection of tables. Its meaning is the conjunction of all of the facts in the tables.

A table is an ordered presentation of a relation.

As a logic program:

\[
\{\{ID=222, Name=\text{\textit{Sally}}\}, \{ID=111, Name=\text{\textit{George}}\}\}
\]

As a relation is a set of facts with the same attributes:

\[
\text{student}(222, \text{\textit{Sally}}), \text{student}(111, \text{\textit{George}})
\]

As a logic formula: student(111, George), student(222, Sally)

Student fact:

\[
\{\{ID=111, Name=\text{\textit{George}}, \text{\textit{Age}}=27\}\}
\]
\{ \ldots \cdot \cdot \cdot \}
\text{Student(SID, Name, String, Age: Int)}

A fuller schema also gives the types of the attributes:

\{\text{Enroll(SID, CID)}\}

\text{Course(CID, Title, Loc)}

\text{Student(SID, Name, Age), Course(CID, Title, Loc)}

\text{particular contents}

\text{The Logical Structure of the Data, apart from its}

\text{Schema of The Database}
Language.

and similar ones. $\text{SQL}$ (to be studied later) is such a possible to a natural language, to express these queries, as close as we would like to have a convenient language.

involve several tables.

Note that some queries involve a single table, and some

Give the names of all students in the History class.

Give the names of all students who are older than 20

For a checkup

Send notices to all employees who had the flu to come

EXAMPLES OF QUESTIONS
THE NEED FOR GOOD DESIGN

...
is has not yet been enrolled in.

We are unable to store the building for a course that

is less obvious at first.

database. This is another type of redundancy, which
Building=NorthLab is written many times into the
The constraint that Course=Chem implies

due to partial updates.

multiple times. There is a danger of inconsistency,
Here we have the name of the building appearing
An item of data should usually appear once only.

PROBLEMS WITH THIS DESIGN
access to the North Lab.

Example: a query that prints the students who need

repilicate information, to speed up the access.

From an efficiency point of view, it might be useful to

AN ADVANTAGE OF THIS DESIGN
The design of relational databases deals with such issues:

\[
\text{Location}(\text{Course, Building})
\]

\[
\text{Enroll(Student, Course)}
\]

\[
\text{BTable(Student, Course, Building)} \text{ into two relations:}
\]

Make the constraints explicit by factoring the schema

**SOLUTION**
Chem North Lab

History

----------

Course Building

Location

Sally History

Chem Sally History

Joe History

----------

Student Course

Enroll
BIRTH

• Health department may prefer to see ACE instead of diseases

Payroll department should see salaries but not

Health department should see diseases but not salaries

See different portions and presentations of the same data.

Various user groups may need to

We may want a design that best reflects the inherent

DIFFERENT USERS NEED DIFFERENT DATA
Some people may need to ask the query: How are salaries correlated?

We need to have flexibility for unanticipated queries.

Let the user see the other information. Let the user see the existing data, without such a view is computed from the existing data, without the health department sees only the derived table:

```
health(Name, Address, Age, Disease)
```

One solution: give each user (class of users) privileges to

```
VIEWS
```
contains this information.

How to change the sum of salaries, if some view

how should this be reflected in the date of birth?

If the user wants to change the age of an employee,

Unfortunately, this is sometimes impossible or difficult.

**VIEWS**

**PROBLEMS WITH UPDATING DATA THROUGH**
the optimization of query processing.

deals with such issues, which are also closely related to

What to do in general? Physical design of databases

sorted by BIRTH

born in a given date range, maybe the file should be

If we want to print the salaries of all the employees

allowing direct access

various SIDs is, maybe the file should be hashed on

If we frequently want to know what the grade for

guarantees efficiency.

The database system should be organized to process

Physical Design
update is secure.

Once the transaction commits, the
all-or-nothing. Once the transaction guarantees that this change is

update the value of Savings in every record.

single "transaction"—execution of a user program—will

is to receive a bonus of 10 in the savings account. A
balance(Savings, Checking).
Suppose each employee
balance(FID, Savings, Checking). Suppose each employee
of employees' accounts in a credit union:

hardware failure. Consider a table describing the balances
The database should remain "consistent" even when

RECOVERY
Consider the following table of employee balances:

<table>
<thead>
<tr>
<th>EID</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>132</td>
</tr>
<tr>
<td>80</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>

System execution of several transactions in a time-sharing environment may also be problematic because of the concurrent execution of several transactions in a time-sharing environment.

**Concurrency**
#101
AVOID: debit from account #121, TT, then credit to

happened before TT
outcome is as if TT happened before TT or TT
Concurrent control must guarantee that the
account #121 to #101 using transaction T2
Concurrently, employee 121 wants to move 40 from
con Currents and print the total of employee balances
Suppose that we are running a transaction TT

CONCURRENcy (CONTR(,D)