Module 1: Overview of Data Management

Fall 2026

Cheriton School of Computer Science

CS 338: Intro to Database Management

Outline

ACME PAYROLL System: A Case Study

ACME Corporation wishes to develop the PAYROLL *information system* (IS) to more carefully manage information about its personnel, their salaries, etc.

Infrastructure for PAYROLL includes:

- A server computer with mass storage,
- A new *payroll application team* (PAT) and
- An existing *database administration* (DBA) group.

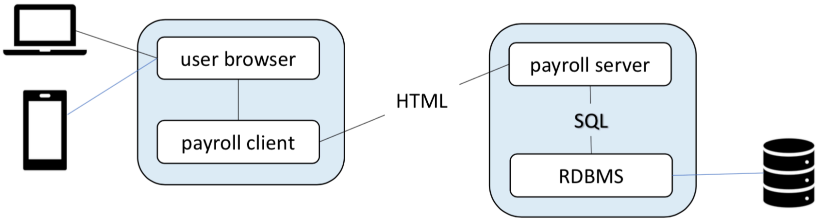
ACME will use an existing *relational database management system* (RDBMS) running on their server computer to manage all PAYROLL information.

PAT will be responsible for the PAYROLL application.

DBA will continue to be responsible for the server computer and for supporting all ISs, including ACME's existing INVENTORY and ONLINE-ORDER systems managed by other teams.

The PAYROLL system will be deployed using a three-tier architecture.

Three-Tier Architecture



PAT develops and maintains the PAYROLL code:

- *payroll client* code (in Javascript) accessed by browsers, and
- *payroll server* code (in C and SQL) running on the server computer.

DBA is also responsible for the RDBMS, which is also used by ACME's existing INVENTORY and ONLINE-ORDER systems.

Information

A DBMS manages two kinds of information: metadata and data.

Some *logical metadata* designed by PAT for PAYROLL:

1. There is a kind of entity called an `employee`.
2. There are attributes called `employee-number`, `name`, and `salary`.
3. Each employee entity has values for attributes `employee-number`, `name`, and `salary` that exceeds $60,000 (USD).
4. Employees are identified by their `employee-number`.

Sample *data*, important to ACME, enabled by the metadata:

5. Mary is an employee.
6. Mary's employee number is 3412.
7. Mary's name is "Smith, Mary".
8. Mary's salary is $92,000 (USD).

Information (cont'd)

A choice of some *physical metadata* specified by DBA for PAYROLL:

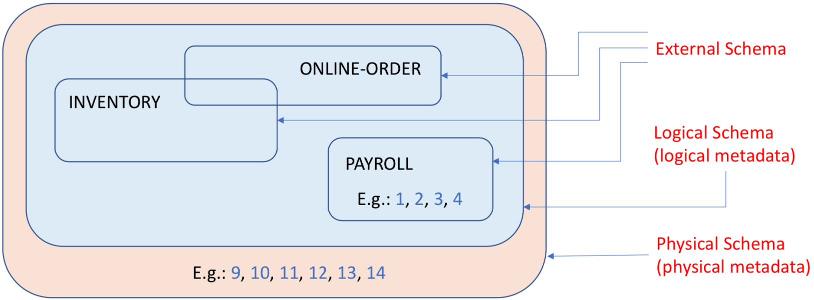
(*concrete data structures on the server computer*)

9. There is a file of records called `emp-file`.
10. There are record fields `emp-num`, `emp-name` and `emp-salary`.
11. Each `emp-file` record has the fields `emp-num`, `emp-name` and `emp-salary`.
12. File `emp-file` is organized as a B-tree data structure that supports an `emp-lookup` operation, given a value for field `emp-num`.

(*mapping rules between logical design and concrete data structures*)

13. Records in file `emp-file` correspond one-to-one with `employee` entities.
14. Record fields in file `emp-file` encode the corresponding attribute values for `employee` entities, for example, `emp-num` encodes an `employee-number`.

Metadata: Three Levels of Abstraction



PAT members are responsible for the PAYROLL external schema design.

DBA members are responsible for all physical design, and to understand *overall workload* of all IS systems (PAYROLL, etc.).

Information Access

Consider the following formulations of a PAYROLL query:

A PAYROLL *user query* specified by PAT:

15. Find the `salary` for any `employee` whose `employee-number` is given by a parameter `p`.

Observe that the query is:

- *independent* of *physical metadata*,
- *independent* of either INVENTORY or ONLINE-ORDER metadata, and
- *declarative*, i.e., has no indication of how to compute the answers.

A *query plan* selected by the RDBMS for the user query:

16. Invoke operation `emp-lookup(p)`, just once, on file `emp-file`. If an `emp-file` record is found, then extract and return the value of field `emp-salary`.

Observe that the query *only* mentions physical metadata, and qualifies as a query plan or *algorithm* for computing the answers.

ACME PAYROLL System: Observations

(reliability) PAYROLL data is important to ACME.

(concurrency) Lots of concurrent interaction by PAYROLL users.

(data integrity) Lots of *integrity constraints* imposed on PAYROLL data by PAYROLL metadata, e.g.:

- employee numbers are *always* unique, and
- employee salaries are at least $60,000 (USD).

(security) Lots of restrictions on who can access and update both PAYROLL metadata and PAYROLL data.

(productivity)

- PAYROLL metadata should be logically data independent of other external schema.
- PAYROLL metadata should be well-designed.
- PAYROLL code should be physically data independent.
- PAYROLL code should be declarative.

Observations (cont'd)

(data is structured) All queries and updates requested by PAYROLL on information managed by the RDBMS is concise.

- No ambiguity about answers to queries.
- No ambiguity about updates to data.

Outline

1. Case Study: The ACME PAYROLL System
2. **Overview of a Database Management System**
3. History
4. Summary

What is a Database

Database

A *large* and *persistent* collection of metadata and data organized in a way that facilitates efficient *retrieval* and *revision*.

Example of metadata: *There are employee entities that have a name and a salary.*

Example of data: *Mary Smith is an employee who earns $92,000 per year.*

Database Management System (DBMS)

A DBMS is a set of programs that implements a data model to manage a database.

Data Model

A *data model* determines the nature of the metadata and how retrieval and revision is expressed.

General Properties of a DBMS

1. A DBMS adopts some data model for managing structured data via an interface with two sub-languages: a DDL, and a DML.

Data Definition Language (DDL)

The part of a DBMS interface enabling clients to manage metadata.

Metadata includes:

- a logical schema, the logical metadata defining how data is understood,
- a physical schema, the physical metadata defining how data is encoded, and
- external schemata comprised of subsets of the logical schema.

Productivity and performance is improved when a data model supports more expressive logical and physical schema.

Data Independence

Physical Data Independence

A property of application code devoid of any mention of the physical schema.

Enables DBA to modify the physical schema without any need to revise any application code.

DBA is free to replace or augment the PAYROLL `emp-file` B-tree without any need for PAT to revise PAYROLL source code.

Logical Data Independence

A property of *pairs* of external schema in which there is no common logical metadata.

Enables respective application teams to work entirely independently.

PAT can work on developing PAYROLL without any need to learn about INVENTORY or ONLINE-ORDER.

General Properties of a DBMS (cont'd)

Data Manipulation Language (DML)

The part of a DBMS interface enabling clients to manage data. Should be

- physically data independent, devoid of any need to mention or understand physical metadata, and *can also be*
- declarative or non-procedural, devoid of any algorithmic details on how data is manipulated.

Productivity is improved when the DML is non-procedural and more expressive.

2. A DBMS supports physical and logical data independence.

3. A DBMS supports concurrent data manipulation.

Reliability and concurrency issues are addressed by supporting transactions, in particular ACID properties for transactions.

Transactions

When multiple applications access the same data, undesirable results can occur.

Consider two concurrent `withdraw` transactions for a hypothetical banking app (part of the database is the value of `AC`):

USER at ATM 1	USER at ATM 2	
`withdraw(AC,1000)`	`withdraw(AC,500)`	
`Bal := getbal(AC)`		(1), for T_1
	`Bal := getbal(AC)`	(2), for T_2
`if (Bal>1000) {`	`if (Bal>500) {`	
`<give-money>`	`<give-money>`	
`setbal(AC,Bal-1000)}`		(3), for T_1
	`setbal(AC,Bal-500)}`	(4), for T_2

AC will be negative if at least \$1,000 but less than \$1,500 to start, and DML requests are executed in the order (1), (2), (3), then (4)!

A DBMS should inform one of the two uses that their `withdraw` transaction has been *denied* due to lack of funds.

Transactions (cont'd)

Transaction

A sequence of *indivisible* DML requests. Applications access a database via transactions.

An application programmer may assume exclusive access to the database within a transaction. The DBMS schedules DML requests from *all* transactions in such a way that guarantees data integrity.

ACID Properties of a Transaction

Atomicity	A transaction occurs entirely, or not at all.
Consistency	Each transaction preserves the consistency of the database.
Isolation	Concurrent transactions do not interfere with each other.
Durability	Once completed, a transaction's changes are permanent.

General Properties of a DBMS (cont'd)

4. A DBMS guarantees data is reliably recorded and can be recovered in case of hardware or software failure.
5. A DBMS provides *access control* to information via *data access permissions* relating to users and roles.
6. A DBMS provides utilities for database monitoring and maintenance.
7. A DBMS supports a variety of users:
   - *IS Users*, e.g., ACME personnel running the PAYROLL app on their desktops, laptops or cell phones.
   - *IS Developers*, e.g., members of PAT.
   - *Database Administration (DBA Group)*, e.g., ACME's DBA group.
   - *Expert Users*, e.g., ACME personnel knowledgable of the PAYROLL external schema and of the DML, and interact directly with the RDBMS.
   - *Implementors*. A DBMS can itself be *extensible*.

Duties of the DBA Group

- Assisting with external schema integration.
- Monitoring and tuning DBMS parameters to optimize performance.
- Management of the physical schema.
- Loading and reformatting the database.
- Management of security and reliability.

The Relational Data Model (RM)

Based on *first order predicate logic*.

Proposed by Ted Codd in 1970.

A watershed achievement is helping to resolve data integrity issues and improve productivity.

Fundamental in codifying the three big ideas underlying a DBMS:

1. physical data independence,
2. data manipulation that is declarative, and
3. interaction via transactions.

SQL is the standard interface for the RM, and features:

1. integrity constraints,
2. a security model,
3. a declarative DML, and
4. a transaction model with ACID properties.

An Example Relational Database

AUTHOR

aid	name
1	Sue
2	John

WROTE

author	publication
1	1
1	4
2	2
1	2

PUBLICATION

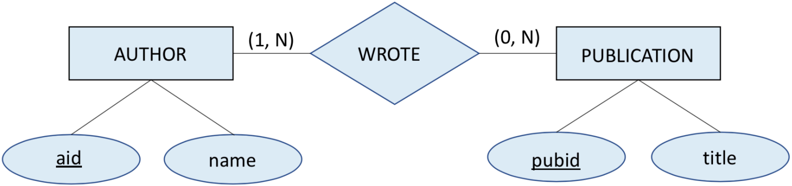
pubid	title
1	Mathematical Logic
3	Trans. on Databases
2	Principles of DB Systems
4	Query Languages

Exercises:

1. Which column(s) serve to identify entities?
2. Which columns have values that must occur in which other columns?
3. What does the following SQL query compute?

```
select distinct name from AUTHOR
where not exists (select * from WROTE where author = aid)
```

The Entity Relationship Data Model (ERM)



Exercises:

1. Which denote sets of entities?
2. Which denote sets of relationships?
3. What does the label “(1, N)” imply?
4. What does the following ERM query compute?

$$\{(a) \mid \text{AUTHOR}(a) \land \neg\exists p.\text{WROTE}(a, p)\}$$

The Nested Relational Model (Nested RM)

PUBLICATION

pubid	title	authors	
		aid	name
1	Mathematical Logic	1	Sue
3	Trans. on Databases		
2	Principles of DB Systems	2 1	John Sue
4	Query Languages	1	Sue

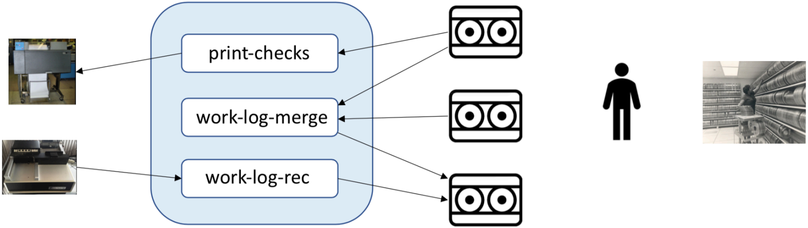
The hierarchical data model is the Nested RM with a navigation DML.

Exercises:

1. What do we get by un-nesting `authors` in `PUBLICATION`?
2. Where might nested relations be useful in a three-tier IS?

Outline

Batch System Architectures: late 1950s – mid 1960s



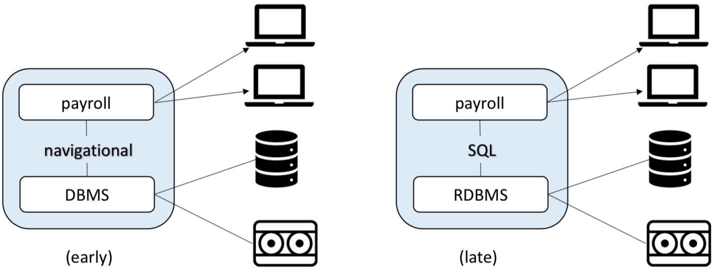
An early PAYROLL system: a collection of batch programs:

- input streamed from tape drives and card readers, and
- output streamed to tape drives and printers.

PAYROLL database: resides on magnetic tapes stored in a tape library.

Data model: hierarchical data model, with a *physically data dependent* navigational DML (JSON!).

Two-Tier Architectures: mid 1960s – present



The PAYROLL system: server and client code is a single process.

PAYROLL database: resides on disk drives (tape drive backup).

Data model:

- Initially, the network data model, which is the relational model limited to a navigational DML.
- By mid-80s, the full relational model.

Advent of RM: 1970s

Edgar Codd proposes relational data model (1970).

- adoption of predicate logic $\rightarrow$ *declarative* query language

Charles Bachman wins ACM Turing award (1973).

- "The Programmer as Navigator"

Peter Chen proposes ER model (1976).

Transaction concepts (Jim Gray and others).

IBM's System R and UC Berkeley's Ingres systems demonstrate feasibility of relational DBMS (late 1970s).

Commercialization of RM: 1980s

Development of commercial relational technology.

- IBM DB2, Oracle, Informix, Sybase

Edgar Codd wins ACM Turing award (1981).

SQL standardization efforts through ANSI and ISO.

Object-oriented DBMSs (late 1980's to middle of 1990's).

- persistent objects
- object id's, methods, inheritence
- navigational interface reminicent of hierarchical model

Evolution of RM: 1990s – present

Continued expansion of SQL and system capabilities.

New application areas.

- the Internet
- On-Line Analytic Processing (OLAP)
- data warehousing
- embedded systems
- multimedia
- XML
- data streams

Jim Gray wins ACM Turing award (1998).

Relational DBMSs incorporate objects (late 1990s).

Many new players in the DB industry (2000+).

Michael Stonebraker wins ACM Turing award (2014).

Outline

Database Systems are Everywhere

Longstanding applications:

- payroll,
- general ledger,
- inventory control,
- banking and financial systems, and
- online reservation systems.

More recent applications:

- online purchasing,
- analytic processing,
- human resources,
- telecommunications,
- social media, and
- navigation.

Summary (cont'd)

Using a DBMS to manage information helps:

- removing common code from applications,
- providing uniform access to data,
- guaranteeing data integrity,
- managing concurrent access,
- protecting against system failure and ensuring reliability, and
- setting and enforcing access policies for data.

Topics for Reflection: Data Models

Capacity for incompleteness, or *not knowing*:

- (disjunctive) Mary's age is either 32 or 33.
- (open world) Not all salaries are known by PAYROLL, or not all employees are known by PAYROLL.
- (statistical) The average salary in sales is $82,000.

Capacity for managing more general kinds of knowledge:

- (temporal) Have salaries ever decreased?
- (geographic) How far is the Toronto office?
- (modal) Employees must eventually retire.

Topics for Reflection: Information Retrieval

ACME's PAT adds the attribute `description` to `employee`.

Example PAYROLL data relating to `description`:[1]

1. Mary's `description` is "`Mary is one of our most productive employees in the more technical areas of the payroll system. Mary gets along extremely well with her colleagues, and is always punctual.`"

ACME needs to express and evaluate the following user query.

2. Find the `name` of each `employee` entity that would be a good choice to lead a project to incorporate recent changes to government regulations on income tax deductions in the PAYROLL system.

[1] `description` values correspond to (possibly long) sequences of characters at one level, and to (unconstrained) English text at another.

How is the choice of data model and user query language affected by this scenario?