Databases: Introduction and overview

Akhil Kumar
Smeal College of Business
Penn State University
(axk41@psu.edu)
Outline

• Examples/Motivation
• Data modeling
• Transaction management
• Next generation databases
• Conclusions
Favorite application – Travel –

The ultimate "Aloha" Experience:

Sabre Reservation system

- FEATURES: Travel services from 400 airlines, 60,000 hotels and 50 car rental companies, railroads, cruise lines; Builds one billion fares, updated 5 times per day; Handles bookings for 35% of all reservations made world-wide amounting to more than $70 Billion annually.

- Average transaction time: less than 3 seconds,
- Peak transaction rate: 13000 transactions per second.
- Annual bookings: more than 300 million.
- Equipment: 31 mainframes, 57 terabytes of storage.
- Processing Power: 11,000 MIPS of processing power.
- Security System: withstand high winds, tornadoes, earthquakes
- Protection: 40 inches of reinforced concrete
- System reliability: 99.998%
Favorite application - Banking
Favorite application – at a bar

• A bar database
  – Swipe driver’s license at bar
  – Look up name, address, birth date, etc.
  – Summarize information
    • On Tuesdays over-40 crowd is large
    • Thursday customers come from upscale zip codes
    • Women make up 80% of crowd when Elvis performs
Data Model

- Consists of concepts and tools for describing
  - Basic or raw data ("price of a chair is 50.99")
  - Data relationships ("ABC Co. supplies chairs")
  - Data semantics ("Unit of price is US $")
  - Data constraints ("A Minimum shipment is 1000")
Entity Relationship Modeling

• Build an E-R model of real world situations
  – Entities (objects or things of interest)
    • E.g. faculty, students, courses, offerings
  – Relationships between entities
    • E.g. Erica is taking Database from Prof. Ball
• Used widely for database design
  – build E-R model and then convert to the relational model
E-R Diagram for manufacturing company

[Aside: In a hotel database, does a reservation belong to guest, room, date?]
An entity is an important "thing" in the application area. A relationship relates two or more entities together.
Cardinality Notation

Inside symbol: minimum cardinality

Outside symbol: maximum cardinality

Single line: one cardinality

Crow’s foot: many cardinality

Circle: zero cardinality
The Three-Schema Architecture

- **External Schema (User View)**
  - view1
  - view2
  - view3

- **Conceptual Schema**

- **Physical Schema**
Three Schema Architecture for Database Development

• External Schema
  – Describes a part of a database that is of interest to a particular user (also called a “view”).
  – Subset of conceptual schema
• Conceptual Schema
  – Describes structure of the entire database (e.g., ER model, relational model)
  – Independent of a specific DBMS
  – No details of physical design
• Physical Schema (or Internal schema)
  – Describes physical structure of the database (tables, indexes, access methods, database distribution.)

Conceptual schema is independent of physical schema - data independence.
Database Development Phases

Data requirements → Conceptual Data Modeling

Conceptual Data Modeling → ERD

ERD → Logical Database Design

Logical Database Design → Tables

Tables → Physical Database Design

Physical Database Design → Internal Schema
Logical data model - Relational Model

In the relational model, all data is stored in relations or tables.

Example Relations (or Tables)

<table>
<thead>
<tr>
<th>Cust_ID</th>
<th>Name</th>
<th>Salesperson</th>
</tr>
</thead>
<tbody>
<tr>
<td>8023</td>
<td>Anderson</td>
<td>Smith</td>
</tr>
<tr>
<td>9167</td>
<td>Bancroft</td>
<td>Hicks</td>
</tr>
<tr>
<td>7924</td>
<td>Hobbs</td>
<td>Smith</td>
</tr>
<tr>
<td>6837</td>
<td>Tucker</td>
<td>Hernandez</td>
</tr>
<tr>
<td>8596</td>
<td>Eckersley</td>
<td>Hicks</td>
</tr>
<tr>
<td>7018</td>
<td>Arnold</td>
<td>Faulb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Salesperson</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>South</td>
</tr>
<tr>
<td>Hicks</td>
<td>West</td>
</tr>
<tr>
<td>Hernandez</td>
<td>East</td>
</tr>
<tr>
<td>Faulb</td>
<td>North</td>
</tr>
</tbody>
</table>

The above relations may be described as follows:
SALES1(Cust_ID, Name, Salesperson)
SPERSON(Salesperson, Region)
### Relational Model

- Example of tabular data in the relational model

<table>
<thead>
<tr>
<th>Customer-id</th>
<th>customer-name</th>
<th>customer-street</th>
<th>customer-city</th>
<th>account-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>192-83-7465</td>
<td>Johnson</td>
<td>Alma</td>
<td>Palo Alto</td>
<td>A-101</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>Smith</td>
<td>North</td>
<td>Rye</td>
<td>A-215</td>
</tr>
<tr>
<td>192-83-7465</td>
<td>Johnson</td>
<td>Alma</td>
<td>Palo Alto</td>
<td>A-201</td>
</tr>
<tr>
<td>321-12-3123</td>
<td>Jones</td>
<td>Main</td>
<td>Harrison</td>
<td>A-217</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>Smith</td>
<td>North</td>
<td>Rye</td>
<td>A-201</td>
</tr>
</tbody>
</table>

**Attributes**
A Sample Relational Database

(a) The *customer* table

<table>
<thead>
<tr>
<th>customer-id</th>
<th>customer-name</th>
<th>customer-street</th>
<th>customer-city</th>
</tr>
</thead>
<tbody>
<tr>
<td>192-83-7465</td>
<td>Johnson</td>
<td>12 Alma St.</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>Smith</td>
<td>4 North St.</td>
<td>Rye</td>
</tr>
<tr>
<td>677-89-9011</td>
<td>Hayes</td>
<td>3 Main St.</td>
<td>Harrison</td>
</tr>
<tr>
<td>182-73-6091</td>
<td>Turner</td>
<td>123 Putnam Ave.</td>
<td>Stamford</td>
</tr>
<tr>
<td>321-12-3123</td>
<td>Jones</td>
<td>100 Main St.</td>
<td>Harrison</td>
</tr>
<tr>
<td>336-66-9999</td>
<td>Lindsay</td>
<td>175 Park Ave.</td>
<td>Pittsfield</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>Smith</td>
<td>72 North St.</td>
<td>Rye</td>
</tr>
</tbody>
</table>

(b) The *account* table

<table>
<thead>
<tr>
<th>account-number</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-101</td>
<td>500</td>
</tr>
<tr>
<td>A-215</td>
<td>700</td>
</tr>
<tr>
<td>A-102</td>
<td>400</td>
</tr>
<tr>
<td>A-305</td>
<td>350</td>
</tr>
<tr>
<td>A-201</td>
<td>900</td>
</tr>
<tr>
<td>A-217</td>
<td>750</td>
</tr>
<tr>
<td>A-222</td>
<td>700</td>
</tr>
</tbody>
</table>

(c) The *depositor* table

<table>
<thead>
<tr>
<th>customer-id</th>
<th>account-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>192-83-7465</td>
<td>A-101</td>
</tr>
<tr>
<td>192-83-7465</td>
<td>A-201</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>A-215</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>A-215</td>
</tr>
<tr>
<td>677-89-9011</td>
<td>A-102</td>
</tr>
<tr>
<td>182-73-6091</td>
<td>A-305</td>
</tr>
<tr>
<td>321-12-3123</td>
<td>A-217</td>
</tr>
<tr>
<td>336-66-9999</td>
<td>A-222</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>A-201</td>
</tr>
</tbody>
</table>
This is a simplified, high-level ER diagram showing one-to-one and one-to-many relationships. Primary keys are in bold, minimum cardinalities are not shown.
Example Queries on Petstore Database

- List all animals with yellow in their color.
- List all dogs with yellow in their color born after 6/1/01.
- List all merchandise for cats with a list price greater than $10.
- List all dogs who are male and registered or who were born before 6/1/01 and are grey in color.
- What is the average sale price of all animals?
- What is the total cost we paid for all animals?
- List the top 10 customers and total amount they spent.
- How many cats are in the animal list?
- Count the number of animals in each category.
- List the CustomerID of everyone who bought something between 4/1/01 and 5/31/01.
- List the first name and phone of every customer who bought something between 4/1/01 and 5/31/01.
- List the last name and phone of anyone who bought a registered white cat between 6/1/01 and 12/31/01.
- Which employee has sold the most items?
Simple SQL Queries

• List all columns for all animals

```
SELECT *
FROM Animal;
```

• List all columns for all animals whose color is yellow.

```
SELECT *
FROM Animal
WHERE (Color LIKE '%yellow%');
```

• List only AnimalId, Category, Breed and Color columns for all animals whose color contains the letters ‘yellow’.

```
SELECT AnimalId, Category, Breed, Color
FROM Animal
WHERE (Color LIKE '%yellow%');
```

The % is a wildcard for any string of characters.
The text inside single quotes is case-sensitive.
Models

• Relational model
• Entity-Relationship model
• Other models:
  – Object-oriented models (such as UML)
  – Semi-structured data models
  – XML Based models
  – (Older models: hierarchical model and network model)
Transaction Management –
Concurrency control and crash recovery

• Transactions must be interleaved, i.e., concurrent
• A database without a concurrency control mechanism will be compromised due to interference between users:
  – Lost update problem, caused by write-write conflict
  – Dirty read, caused by read-write conflict
• A real world databases must allow proper concurrent access
## Lost Update Problem - Airlines

<table>
<thead>
<tr>
<th>Transaction A</th>
<th>Time</th>
<th>Transaction B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read SR (10)</td>
<td>$T_1$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_2$</td>
<td>Read SR (10)</td>
</tr>
<tr>
<td>If SR &gt; 0 then</td>
<td>$T_3$</td>
<td></td>
</tr>
<tr>
<td>SR = SR - 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write SR (9)</td>
<td>$T_4$</td>
<td>If SR &gt; 0 then</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR = SR - 1</td>
</tr>
<tr>
<td></td>
<td>$T_5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T_6$</td>
<td>Write SR (9)</td>
</tr>
</tbody>
</table>

**SR: Seats Remaining**
Lost Update Problem: Banking

Husband and wife to take money from the same account with $1000

Husband H
- read $1000
- withdraw $100
- write $900

Wife W
- READ $1000
- Withdraw $200
- WRITE $800

Possible sequence without concurrency control:
1. H read $1000
2. W READ $1000
3. H write $900
4. W WRITE $800

Balance is $900 instead of the correct value of $700. One update is lost.
**Uncommitted Dependency or Dirty Read Problem**

<table>
<thead>
<tr>
<th>Transaction A</th>
<th>Time</th>
<th>Transaction B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read SR (10)</td>
<td>$T_1$</td>
<td></td>
</tr>
<tr>
<td>SR = SR - 1</td>
<td>$T_2$</td>
<td></td>
</tr>
<tr>
<td>Write SR (9)</td>
<td>$T_3$</td>
<td>Read SR (9)</td>
</tr>
<tr>
<td>Rollback</td>
<td>$T_4$, $T_5$</td>
<td></td>
</tr>
</tbody>
</table>

SR: Seats Remaining
Locking Fundamentals

• Locking is the main tool of concurrency control

• Rule: Obtain lock before accessing a data item and release lock at end of transaction

• Wait if a conflicting lock is held
  – Shared lock (S-Lock): conflicts with exclusive locks
  – Exclusive lock (X-Lock): conflicts with all other kinds of locks

• Concurrency control manager maintains the lock table
Crash................ recovery

• Must keep a log:
  – *When Ti writes an object*: the old value and the new value.
    • Log record transferred to disk *before* the changed page!
    – *Ti commits/aborts*: a log record indicating this action.

• Log records are chained together by Transaction id, so it’s easy to undo a specific Transaction

• Log is often *duplexed* and *archived* on “stable” storage.

[Ti: is a transaction running in the database]
Next generation databases

Features:

• Sources of data are widespread and their number is potentially infinite

• The sources vary in nature from being unstructured text on the Web to semi-structured data such as articles, reports and news briefs, to structured sources like conventional database systems

• Application areas may vary from business data to scientific (astronomy, biology, etc.) and geospatial data

• Data may also vary across the media spectrum (from plain text, to sound, graphics and pictures) and also across languages
New applications – interesting problems

• Digitize 50 years of BBC coverage and serve on demand
  Random access to 1 petabyte of data \((10^{18} \text{ bytes})\)
• GIS applications ("alert me if there is cancellation for a
  show/game and I am within 1 mile)
• Analyzing click stream data, continuous stream data
• Voice to natural language to structured data
• Bio-informatics (analyzing large biological data sets)
  – Storing large numbers of gene sequences
  – Finding gene locations and querying gene sequences
  – Learn structure and functions of proteins
Conclusions

• Database systems have come a long way from humble beginnings as simple file systems
• Discussed basic principles of database systems
• Mentioned lots of potential new applications