CS 5/7330

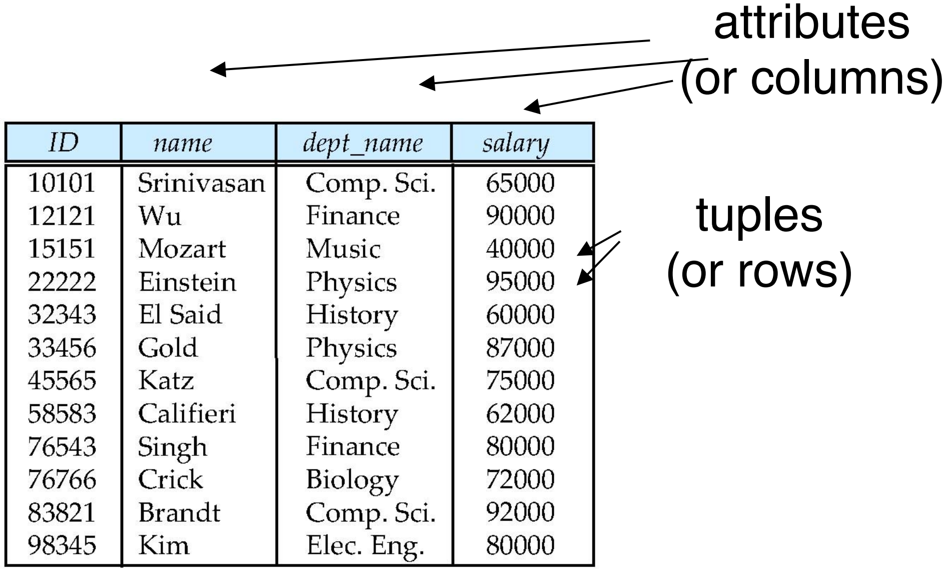
Review / Intro to NoSQL

Database Systems

- Why database systems?
  - Provide means for user to manage data
  - Allow users to
    - Specify data to be stored (data modeling)
    - Specify operation on the data (query language)
    - Ensure consistency and integrity of data (integrity checking)
    - Manage how data is being stored (indexing, file organization)
    - Manage how data is actually retrieved (query processing, optimization)
    - Manage how data is being shared/not shared (concurrency control)
    - Recover data after a failure (recovery)
    - Ensure data are not accessible by people who should not (security)
    - Access data from a variety of sources/locations (distribution processing)

Relational Model

- Data are represented as tuples in relations
- Represented as tables
  - Rows (tuples): each unit of data
  - Columns (attributes) : attributes of each unit



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

attributes
(or columns)

tuples
(or rows)

Relational Model

- Relation can be viewed as SETS of attributes (set in a mathematical sense)
- Constraints on the table
  - First Normal Form
    - But NULL values allowed
  - No duplicate tuples
    - Thus (primary) keys
  - Domain values
  - Referential Integrity

    [ɪnˈtegrəti] n. 正直，诚实；完整，完全；职业操守；（电子数据的）集成度
    - Foreign keys

SQL

- A typical SQL query has the form:

  select $A_1, A_2, \ldots, A_n$
  from $r_1, r_2, \ldots, r_m$
  where P

  - A_i represents an attribute
  - R_i represents a relation
  - P is a predicate.
- The result of an SQL query is a relation.

  ['kwɪri] n. 疑问，询问；问号 v. 质疑，
  对……表示疑问；询问，提问

- API available for a variety of programming languages to interact (e.g. ODBC, JDBC)

  Open Database Connectivity (ODBC) is an open standard application programming interface (API) that allows application programmers to access any database.

  ıva Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a ıtabase. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporatior

- Also database specific APIs are available

Relational Algebra

- An abstract query language on relations
- A set of operations on a relations, returning another relation
- Basic operations:
  - Selection: $\sigma_{condition}$ (R)
    - Pick tuples from the relation based on a condition, keeping all attributes
  - Projection: $\prod_{\text{attribute list}} (R)$
    - Select all tuples, but only keep attributes on the attribute list
  - Set operations: $\cap$,$\cup$, -
    - There are corresponding SQL commands for these (not often used)

Relational Algebra

- Basic operations:
  - Cartesian Product: r x s:
    - Create a table such that every pair of tuples in r, s is match to a topic
    - Basis of merging multiple tables
  - Join: r $\bowtie$ s
    - Cartesian Product followed by a selection
    - The selection usually (not always) match the corresponding attributes (foreign keys) of the two tables.
    - Most common operation for merging tables.

Relational Algebra and SQL

- A typical SQL query has the form:

  select $A_1, A_2, \ldots, A_n$
  from $r_1, r_2, \ldots, r_m$
  where P

Projection in relational algebra

Cartesian Product in relational algebra

Selection in relational algebra
(remember, join = Cartesian
product + selection)

  - A_i represents an attribute
  - R_i represents a relation
  - P is a predicate.
- The result of an SQL query is a relation.

Relational Algebra -- Extension

- Each query input is a table (or set of tables)
- Each query output is a table.
- All data in the output table appears in one of the input tables
- There are many things that relational algebra (and SQL) cannot do
  - E.g. things that require recursion
- Extension of relational algebra to incorporate SQL statements such as
  - SUM
  - AVG
  - MAX
  - MIN
  - GROUP BY…HAVING

Relational algebra – Why?

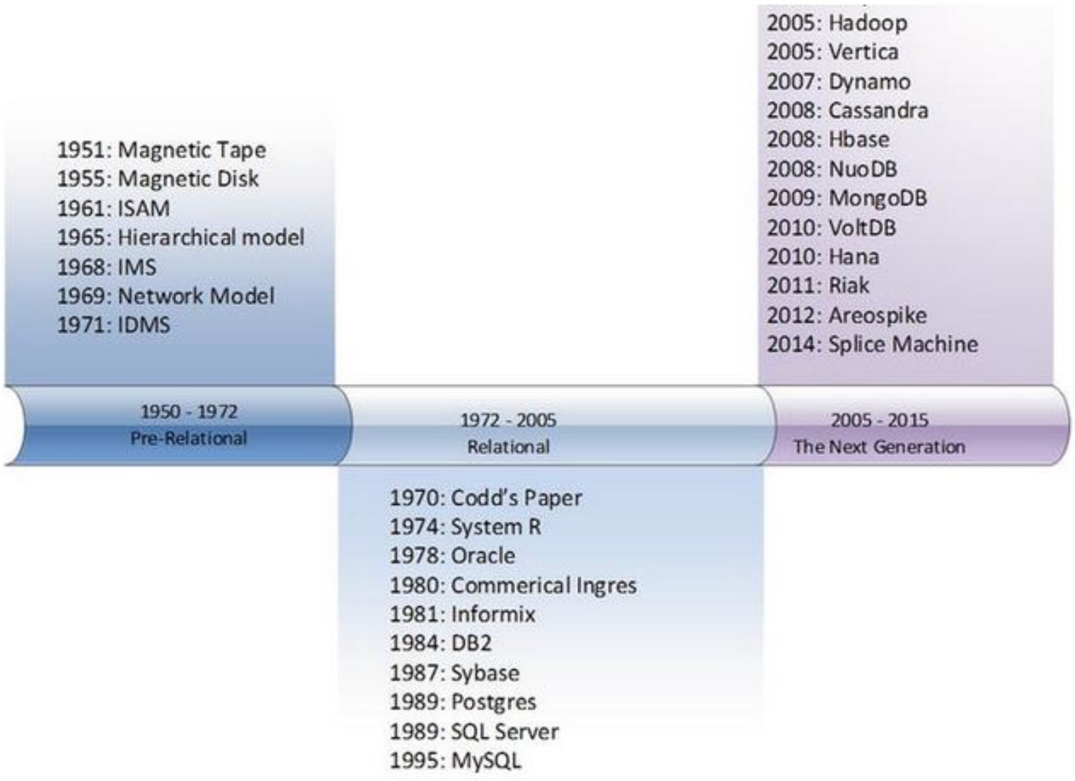
- Very straight forward way of converting SQL statements into a list of operations
- Now executing a query becomes executing a program of such operations
- Implementing them correctly and efficiently will ensure the database system perform well

Course outline

- Outline of the course
  - NoSQL (from data modelling perspective)
  - Internals of a DBMS
    - Query execution and optimization
    - Concurrency Control
    - Recovery
  - Distributed databases / Big Data
    - NoSQL (from a performance perspective)

NoSQL databases

- A history of database systems



1951: Magnetic Tape
1955: Magnetic Disk
1961: ISAM
1965: Hierarchical model
1968: IMS
1969: Network Model
1971: IDMS

1950 - 1972
Pre-Relational

1972 - 2005
Relational

1970: Codd's Paper
1974: System R
1978: Oracle
1980: Commerical Ingres
1981: Informix
1984: DB2
1987: Sybase
1989: Postgres
1989: SQL Server
1995: MySQL

2005 - 2015
The Next Generation

2005: Hadoop
2005: Vertica
2007: Dynamo
2008: Cassandra
2008: Hbase
2008: NuoDB
2009: MongoDB
2010: VoltDB
2010: Hana
2011: Riak
2012: Areospike
2014: Splice Machine

Figure 1-1. Timeline of major database releases and innovations

From: Guy Harrison, "Next generation databases : NoSQL and Big Data", Apress, 2015

First Generation Databases

- Network Model/Hierarchical Model
- Works on mainframes
- Navigation based (i.e. you need to tell the DBMS “where” the data is)
    - E.g. following pointers and links (parent-child etc.)
- Drawbacks
    - Inflexible schema structure (next to impossible to change mid-stream)
    - Navigation based means complex query equal to complex program
        - User have to specify how to get to the data

Second Generation Database

- Relational Model
- Advantages
    - Well-defined mathematical background
    - Normal forms
        - “all non-key attributes must be dependent on “the key, the whole key, and nothing but the key—So Help Me Codd”
    - Separation of physical and logical layers
        - Make things like optimization possible/manageable
    - Full transaction model (ACID)
        - For concurrency and recovery
    - Well suited for client-server systems

Limit of relational models

- First normal form
- No compound value (set/list/sequence etc.)
- To store this require multiple tuples:

Student ID	Course
1	CS 7330
1	CS 7445
2	CS 7330
2	CS 7555
2	CS 7688

Limit of relational model

- Modeling of (undirected) graph can be very tricky

Edge ID	Node 1	Node2
1	1	2
2	1	4
3	3	2
4	5	1
5	4	6

- Do we need to duplicate edge?
- Either query will be complicated (e.g. node1 = 3 or node2 = 3)
- Or inconsistency may occur (deleted (1, 3) but forget to delete (3,1)

Limitations of relational model

- Development of new applications
    - Large amount of data
    - Not necessarily structured
        - Potentially evolving
    - Required high availability and fault tolerance
        - ACID may not be the best options for transactions

NoSQL databases

- More flexible data models
  - Semi-structured / Non-structured
  - Allow for evolving (non-fixed schema)
- Weak consistency model
  - Not requiring ACID properties
- Different storage management
  - E.g. storing via columns
- Designed to allow replication
  - Need to handle consistency (or allow limited level of inconsistency)
- Targeted for Big Data / Web applications

NoSQL Data Models

- Key-Value stores
- Wide Columns stores
- Document stores
- Graph stores