Introduction to Databases

Second La Serena School for Data Science:
Applied Tools for Astronomy
August 2014

Mauro San Martín
msmartin@userena.cl
Universidad de La Serena

Contents

- Introduction

  Databases and scientific data management

- Part I. Relational Databases

  Defining, storing, updating and querying data

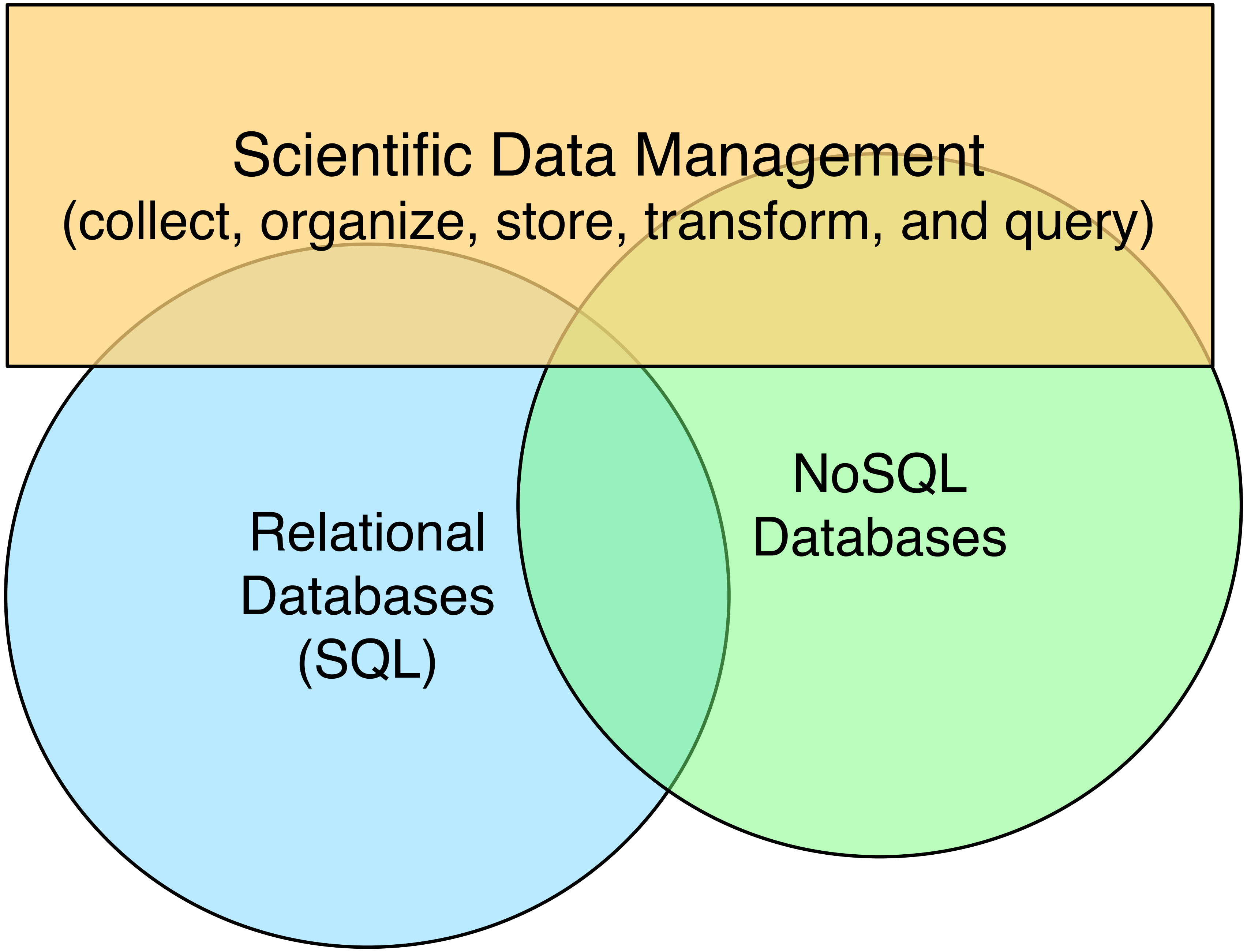
- Part II. No Relational Databases

  Short introduction to alternatives to relational databases

Definitions

- Database
  - An organized and self-describing collection of data, with a intended meaning, and maintained with a purpose.
- Database Management Systems (DBMS)
  - Software system designed and implemented to define, maintain and share a database, and
  - to separate app. logic from low level data I/O.

Scientific Data Management


Scientific Data Management
(collect, organize, store, transform, and query)
Relational Databases (SQL)
NoSQL Databases

Part I.
Relational Databases
Theory

RDBs at a glance

- E. F. Codd 1970

  "A Relational Model of Data for Large Shared Data Banks"

- Main characteristics
  - One simple data structure: relation
  - Solid mathematical foundations
  - Several comprehensive implementations available:

    PostgreSQL, MySQL, Oracle, SQL Server, etc.

- Industry standard since the 80's

Modeling data

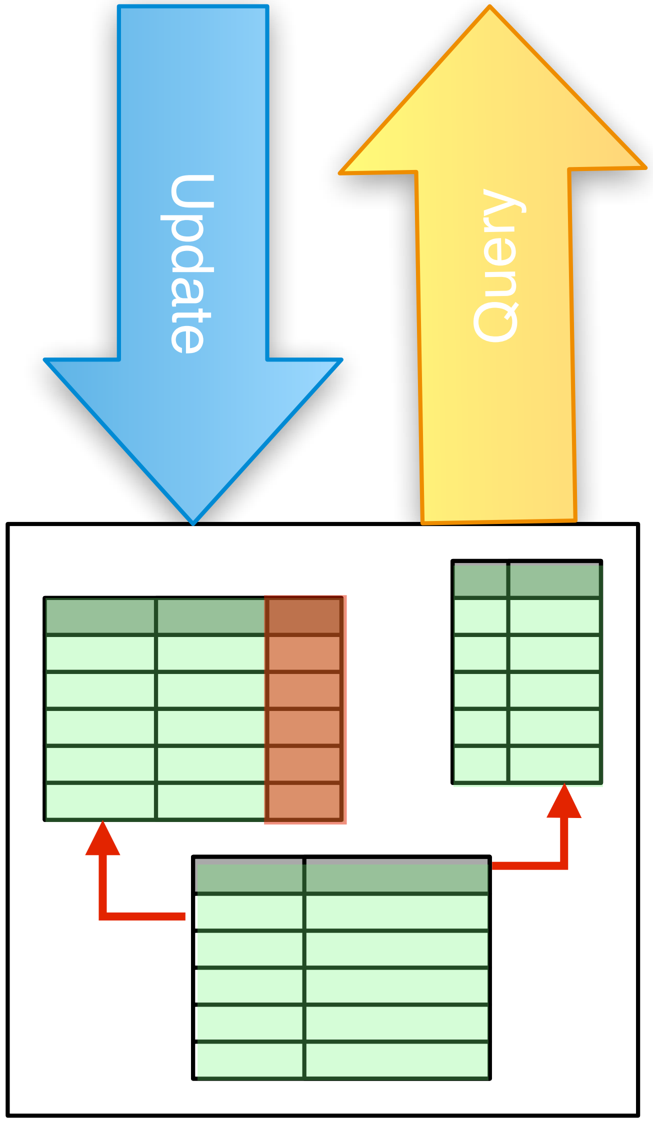
Capturing the world (or the universe)

- The relational data model
  - data structure

    relations/tables: collections of tuples
  - operations (query + update)

    relational algebra and calculus/SQL
  - integrity constraints

    Data type, not null, referential integrity



Schemas

- Schema

  Definition of relations (columns, types, and keys) and integrity constraints.

- Good Schema avoids

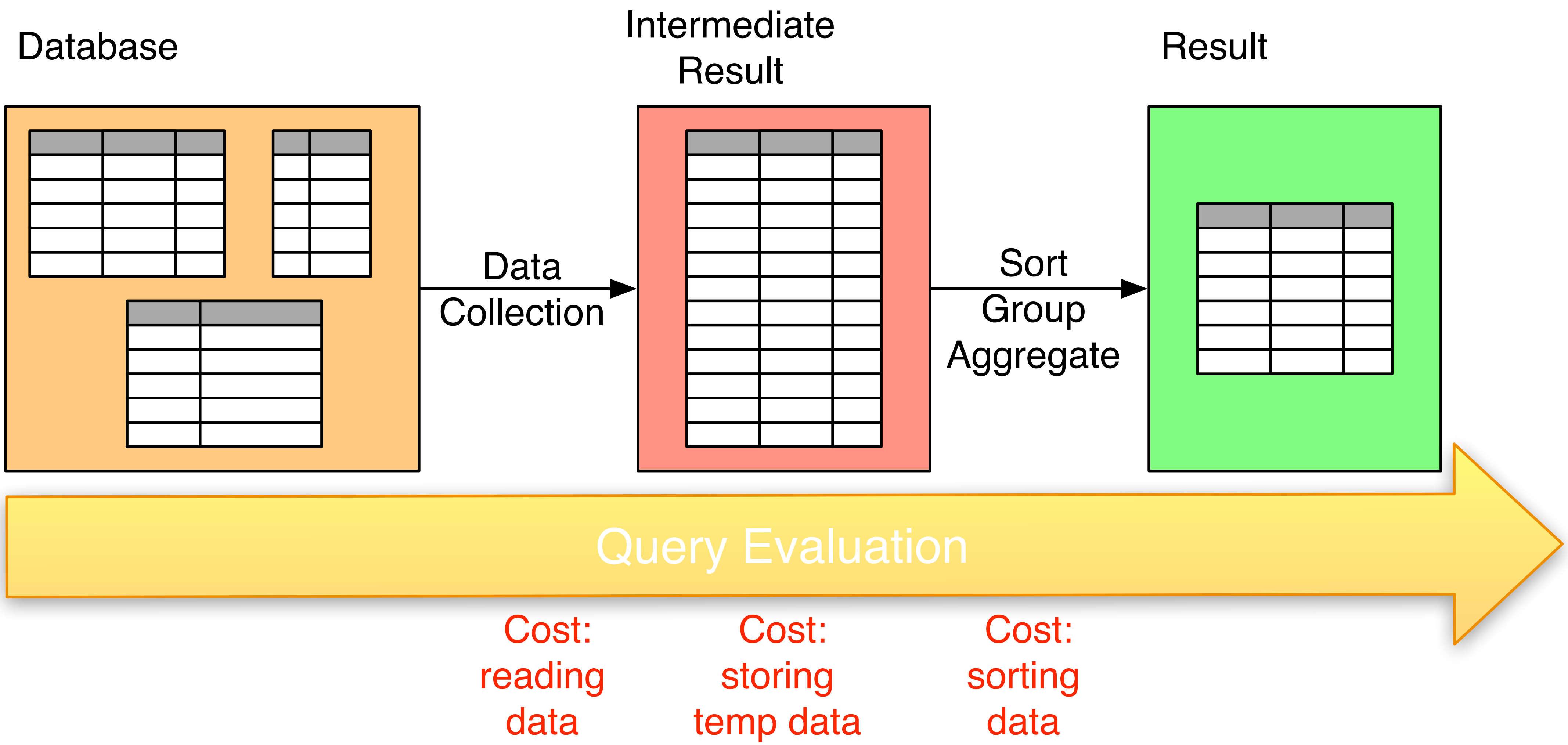
  Data duplication, null values, and update anomalies.

- Normalization: algorithm to build good schemas.

  Identify keys and divide relations (separate columns) with problems.

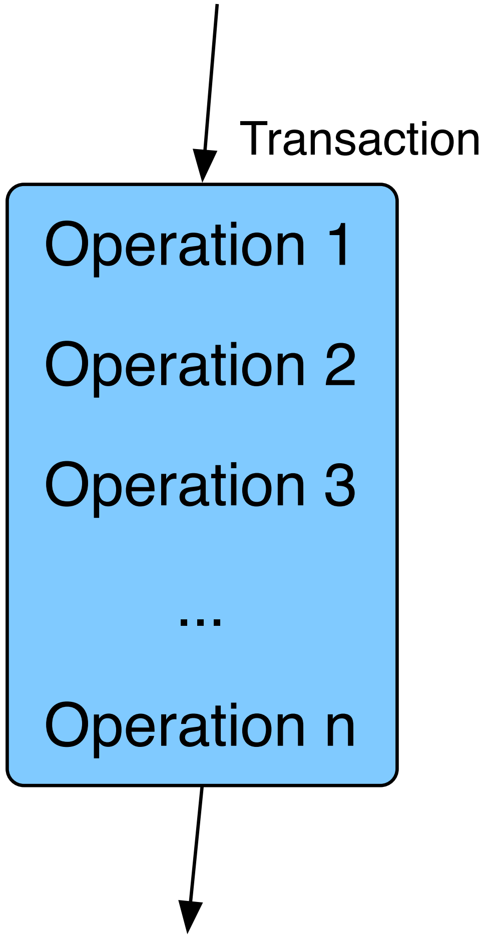
Querying the DB

Map data from DB to the information needed



Updates

- Update: add and modify data.
  - Updates may render the database inconsistent
- Transactions and ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability

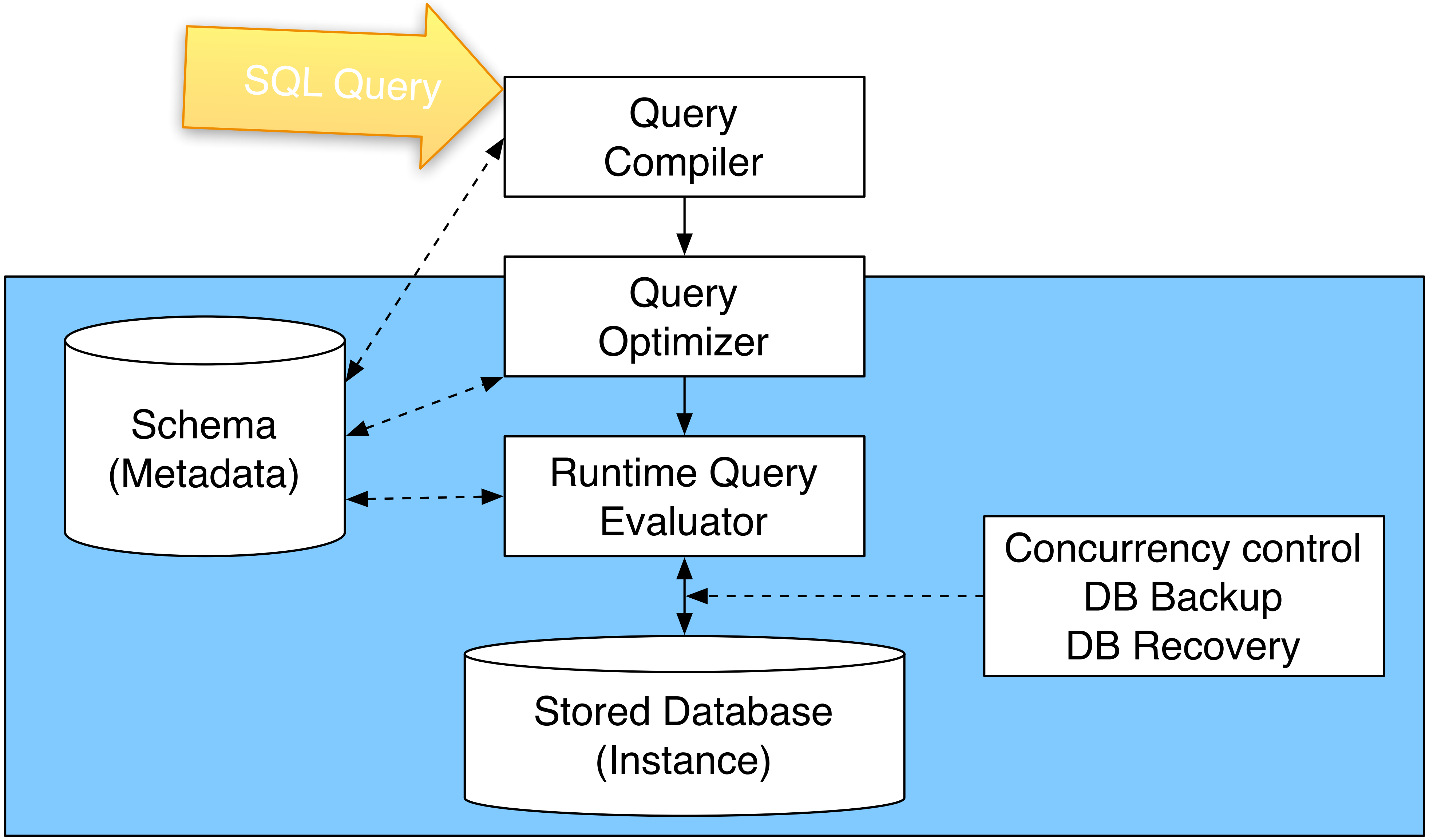


Part I.
Relational Databases

Practice

What is an RDBMS?

A DataBase Management System (software) for the Relational model



RDBMS Objects

- Tables

  Represent data: collection of records

  Record: set of attributes (columns)

ObjectID	A	B
ID1	3.4	a
ID2	4.0	b
ID2	2.1	c

- Views: named queries
- Indices: improve search and access time
- Functions: extend query language

SQL

- Structured Query Language
- Actually it includes
  - Data Definition Language (Schema)

    ```
    create table myTable(number int, letter char)
    ```

  - Data Manipulation Language (Update)

    ```
    insert into myTable values(1, ‘a’)
    ```

Querying the DB

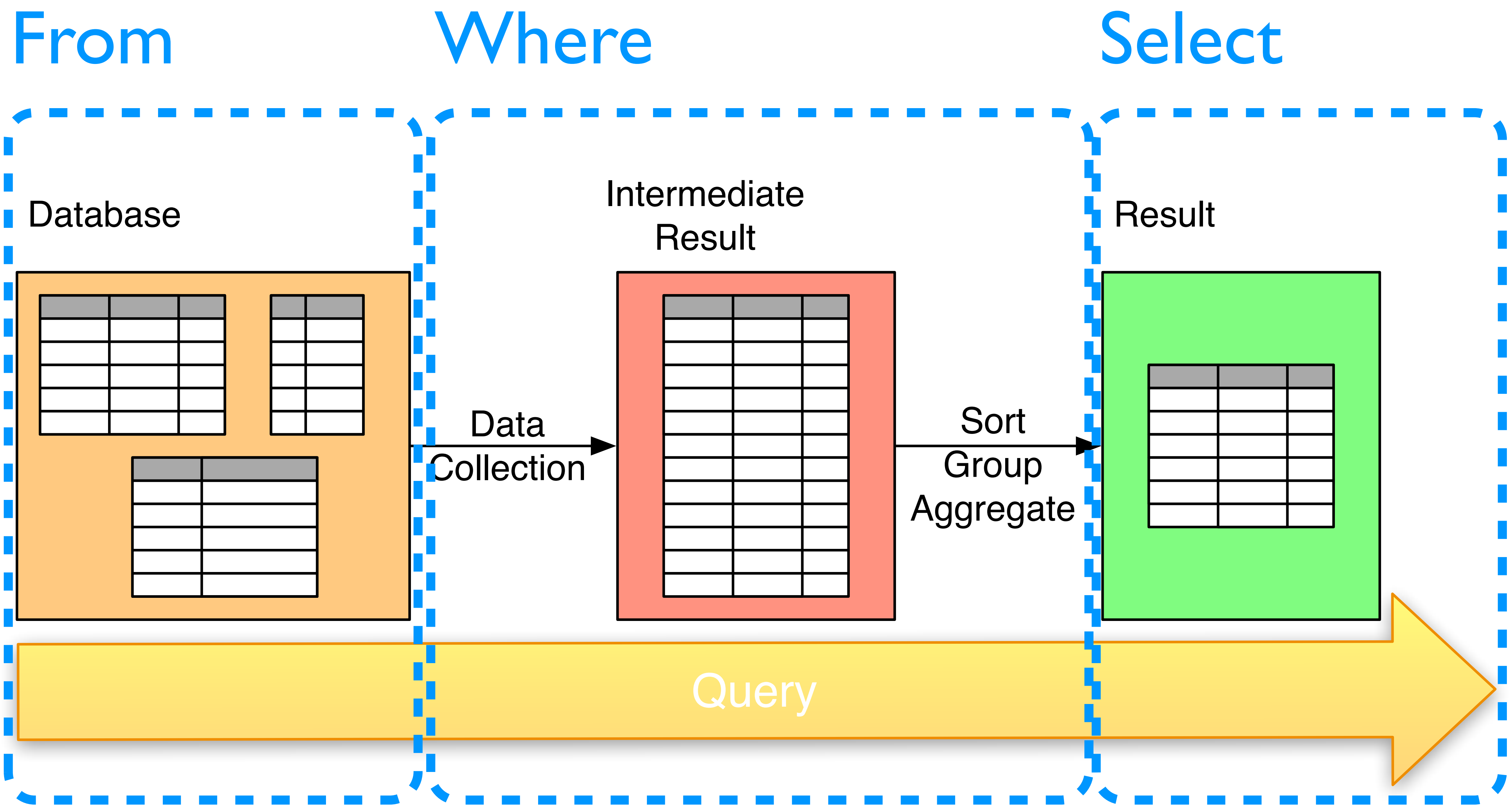
- Basic Query Structure

  SELECT: definition of the output table (set of columns)
  FROM: identification of source tables
  WHERE: optional condition (filter or join)

- Aditional blocks

  GROUP BY: group defining criteria
  HAVING: optional condition un aggregate values
  ORDER BY: sorting criteria for the result

Query Evaluation



Note that query results are relations (query composition)

Executing Queries

- Parametric
- SQL
  - System console
  - Applications and web interfaces
- From code
  - Parametric from programmer's perspective
  - Languages + libraries

Query Examples

- Example Database
  - Source: SLOAN DR10
- Schema
  - Tables

```
photoObj(oid, ra, dec, g, r)
specObj(oid, class, subclass)
```

Basic Query

```
SELECT * FROM photoObj;
```

```
SELECT oid, class
FROM specObj
WHERE class = ‘GALAXY’;
```

Complex Conditions

```
SELECT oid, ra, dec
FROM photoObj
WHERE
g < 12
and r < 12
and g - r < 0;
```

Joins

```
SELECT p.oid, p.ra, p.dec, s.subclass
FROM photoObj as p, specObj as s
WHERE
p.oid = s.oid
and p.g < 12 and p.r < 12
and p.g - p.r < 0
and s.class = ‘GALAXY’;
```

Groups and Aggregates

```
SELECT s.subclass, count(*)
FROM photoObj as p, specObj as s
WHERE
p.oid = s.oid and p.g < 12
and p.r < 12 and p.g - p.r < 0
and s.class = ‘GALAXY’
GROUP BY s.subclass;
```

Sub-Queries

```
SELECT oid, ra, dec
FROM photoObj
WHERE g < 12 and r < 12
and g - r < 0
and oid in(SELECT oid
            FROM specObj
            WHEREs.class = ‘GALAXY’);
```

Query Complexity

- Data Volume
  - I/O based cost model

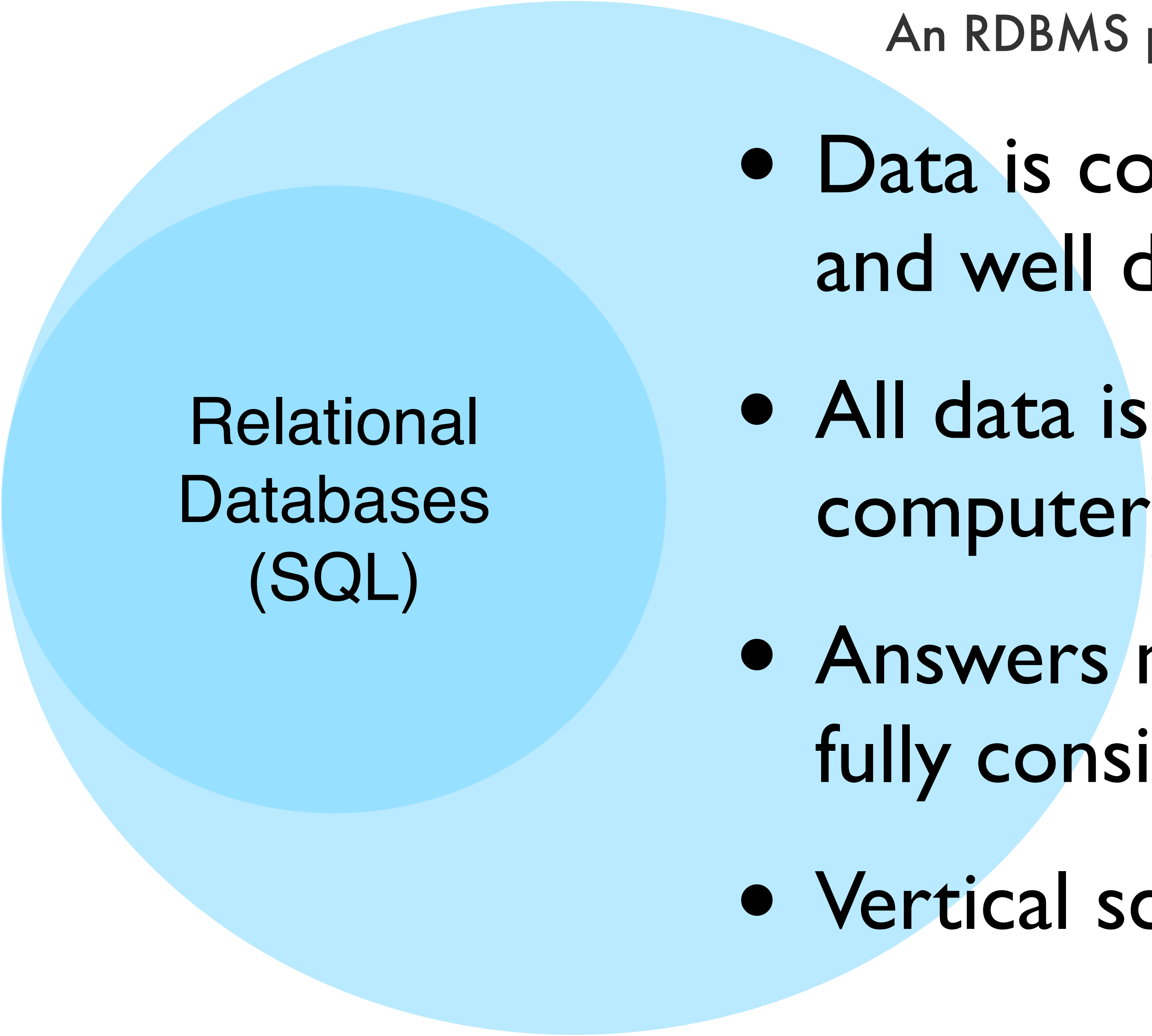
    number of reads from and writes to persistent storage
- Query Complexity
  - table size: n, number of tables: k
  - search: O(1) to O(log n) to O(n)
  - joins: O(n) to O(n^k)
  - sort, group, and aggregates: O(n log n)

    (size of intermediate result)
  - subqueries: hard for the optimizer

Part II.
NoSQL

Not only SQL

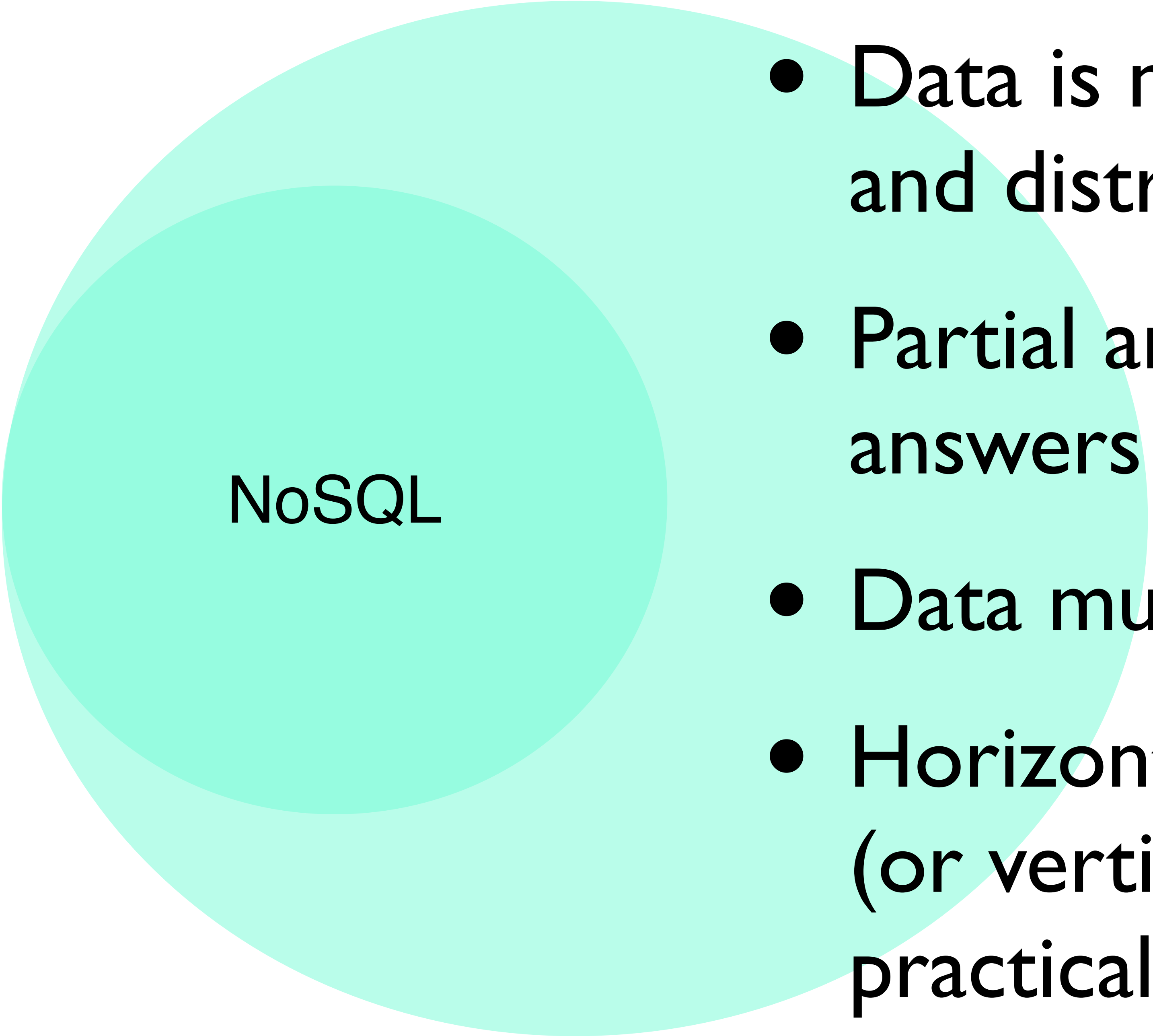
RDBMS Comfort Zone



An RDBMS performs better when ...

- Data is complete, homogeneous and well defined.
- All data is together (in the same computer).
- Answers must be complete and fully consistent.
- Vertical scaling is possible.

NoSQL Comfort Zone



- Data is massive, heterogeneous, and distributed.
- Partial and eventually consistent answers are acceptable.
- Data must be always available.
- Horizontal scaling is preferred (or vertical scaling is not practical).

NoSQL Databases

- Aggregate

  Key: identify each aggregate

  Data: heterogeneous collections of attributes as name/value pairs.

- Main Types
  - Key-Value Stores

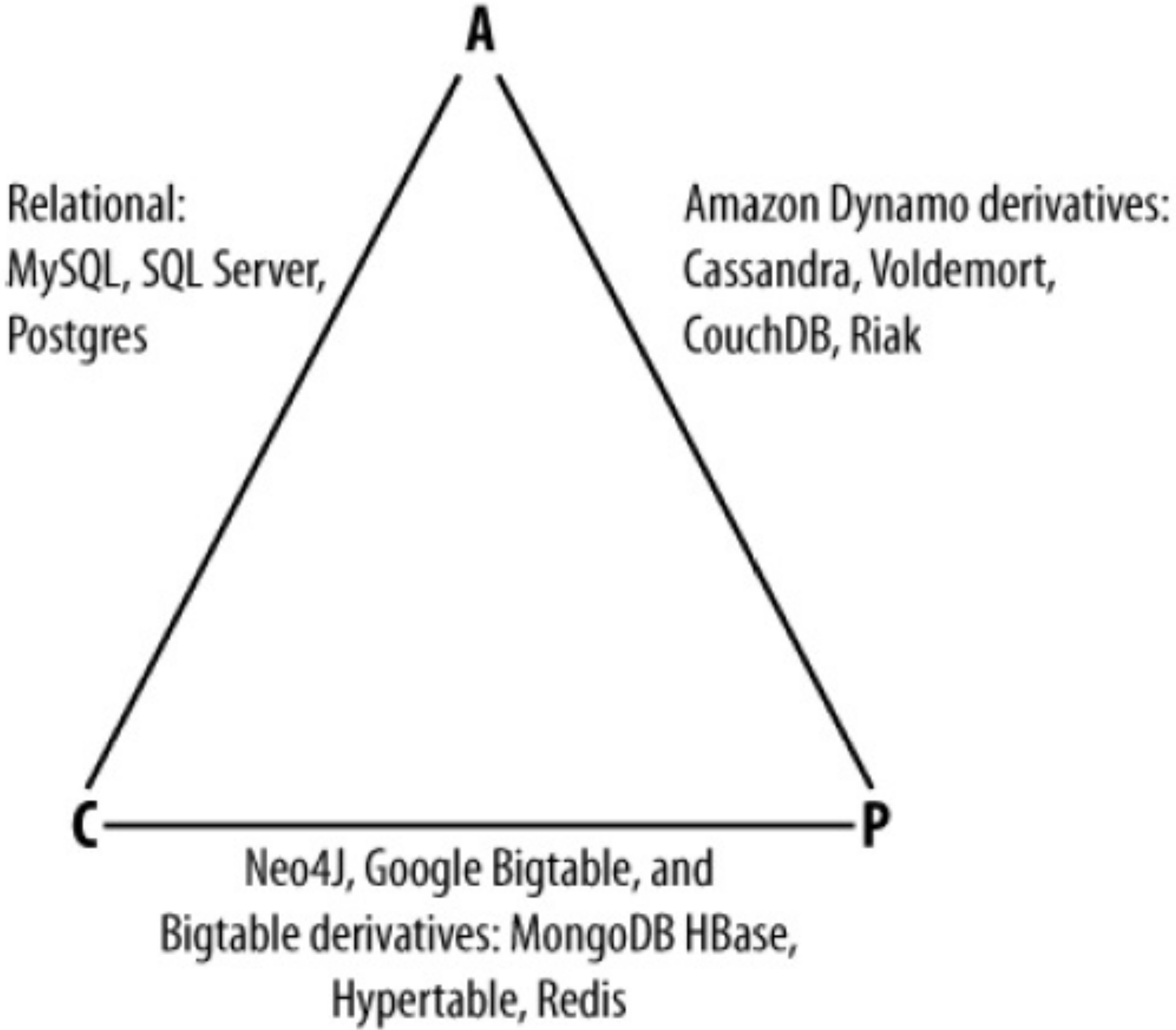
    fast to retrieve data with unknown structure
  - Document Databases

    (mostly) tree structured data
  - Column-Family Stores

    complex structured data

CAP *Theorem*

- Consistency
- Availability
- Partition Tolerance



Choose two!

Query Evaluation

- Map-Reduce

  Parallel (cluster) data-processing pattern.

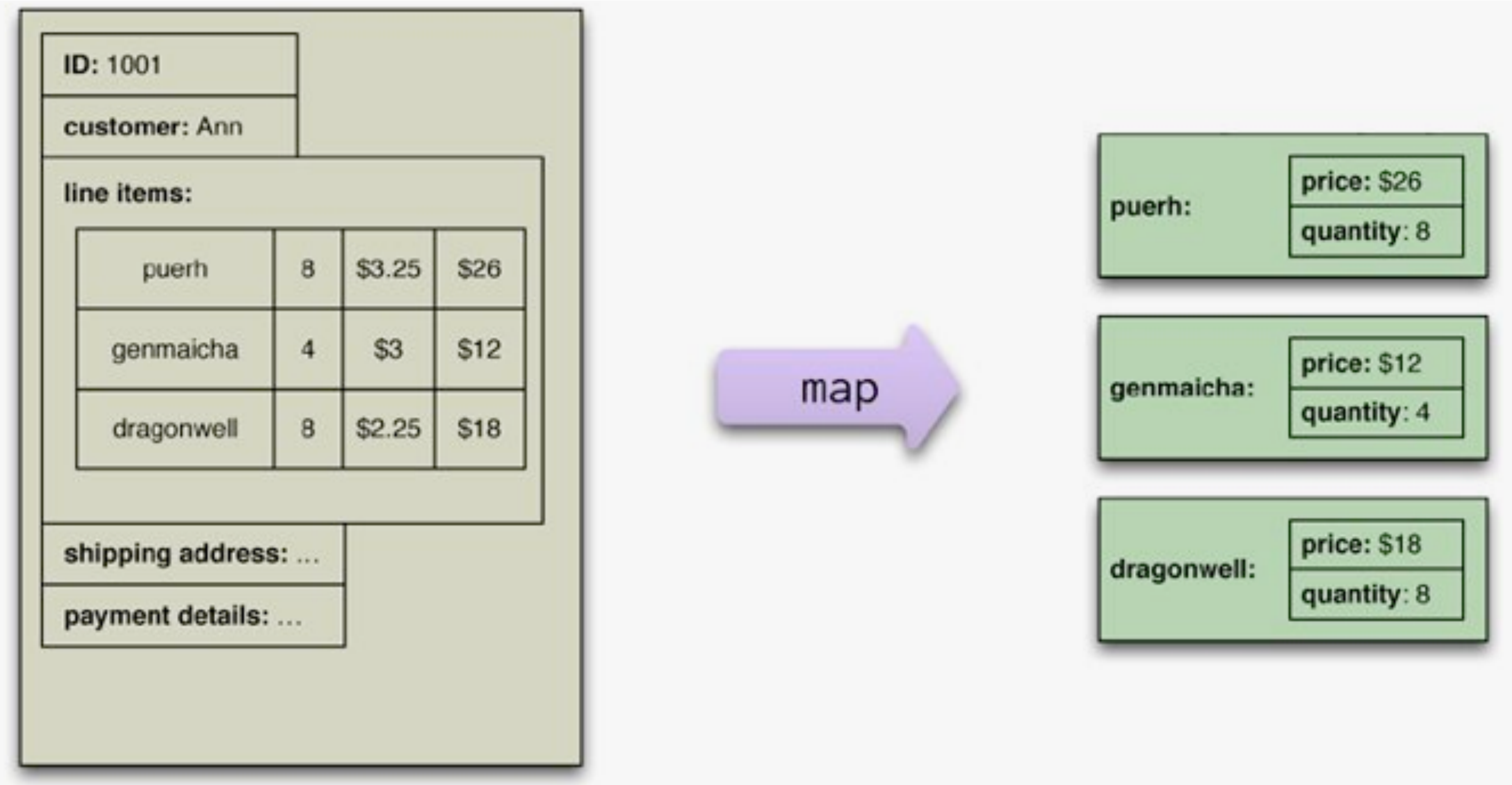
- Two steps
  - Map

    Input is an aggregate, output is a bunch of key-value pairs.

    Each map is independent (across aggregates in all the cluster).
  - Reduce

    Map results are collected, sorted and combined.

Map-Reduce: Map



ID: 1001

customer: Ann

line items:

puerh	8	$3.25	$26
genmaicha	4	$3	$12
dragonwell	8	$2.25	$18

shipping address: ...

payment details: ...

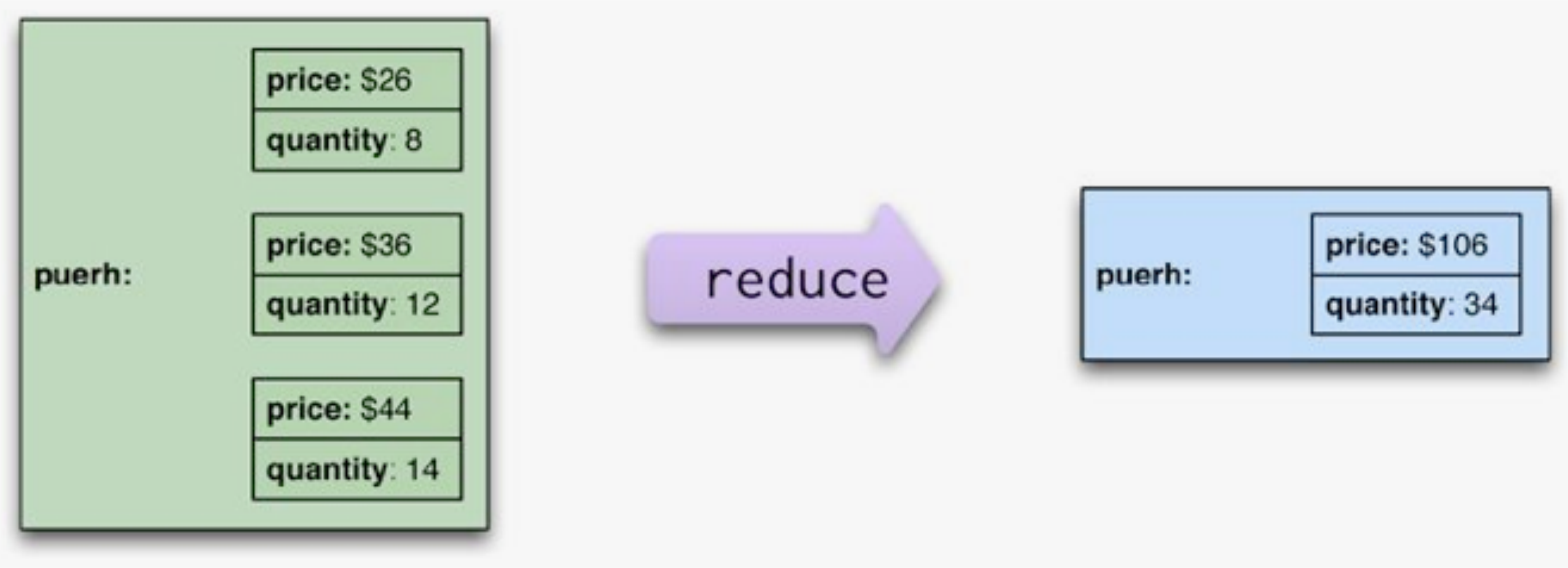
puerh: price: $26 quantity: 8

genmaicha: price: $12 quantity: 4

dragonwell: price: $18 quantity: 8

Fuente: Fowler and Sadalage, NOSQL Distilled.

Map-Reduce: Reduce



Fuente: Fowler and Sadalage, NOSQL Distilled.

Summary

- RDBMS
  - Tables: collections of records with keys
  - SQL

    Queries: basic, join, groups and aggregates, subqueries.
- NoSQL
  - Aggregates: collections of key-value pairs with one identifier.
  - Map-Reduce