Database basics

Things you'll learn, probably forget and hopefully remember again

May 2018

Table of content

Learn @ uni -> forget -> refresh -> grok @ work

- ✓ Data modelling
- ✓ Data integrity (ACID)
- ✓ Big O, indexes, joins, sorts & query optimisation
- ✓ SQL

Goodies you'll learn @ work

Not-so-goodies you'll learn @ work

About me

Architect/Engineer at PayPal Singapore Development Center.

Worked around APAC - from small NZ startups to large multi-nationals.

Technically specalised in back-end: databases, APIs, machine learning.

Loves building useful products.

Data Modelling

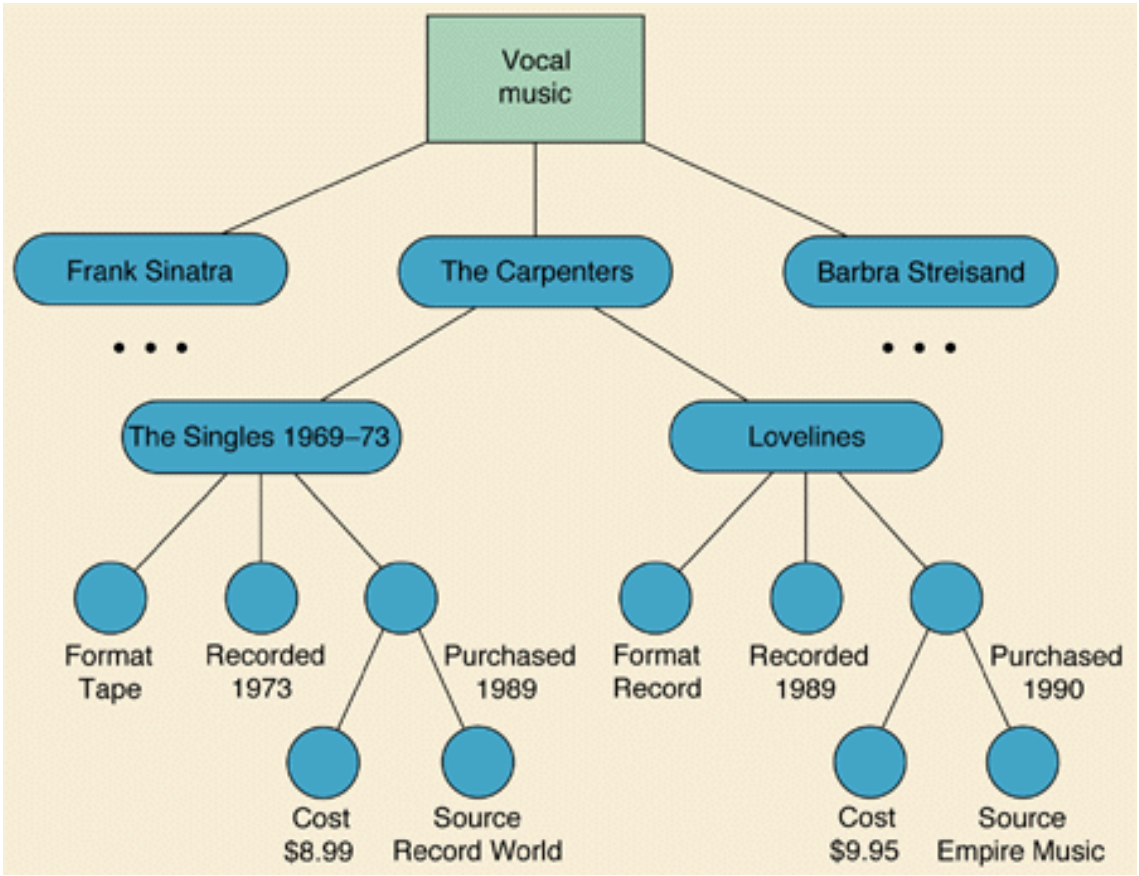
Hierarchical databases

"A hierarchical database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links."

"The hierarchical structure is used primarily today for storing geographic information and file systems."

Big problem: duplication in many-to-many relationships

Source: https://en.wikipedia.org/wiki/Hierarchical_database_model

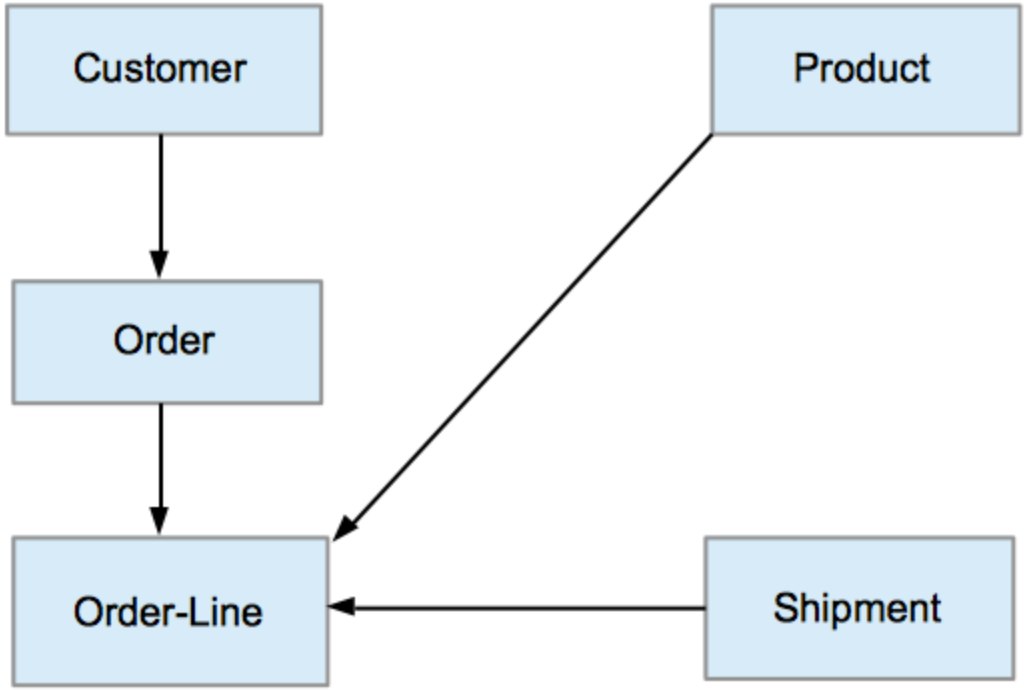


Network databases

"Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice."

- Fast
- Low-level access
- Replaced by relational databases

Source: https://en.wikipedia.org/wiki/Network_model



Relational databases

Relational algebra

- Set theory (union, intersect, minus…)
- Joins (Cartesian, natural, semi, outer, anti…)
- Aggregation

"first created by Edgar F. Codd while at IBM, is a family of algebras with a well-founded semantics used for modelling the data stored in relational databases, and defining queries on it."

Source: https://en.wikipedia.org/wiki/Relational_algebra

Relational model

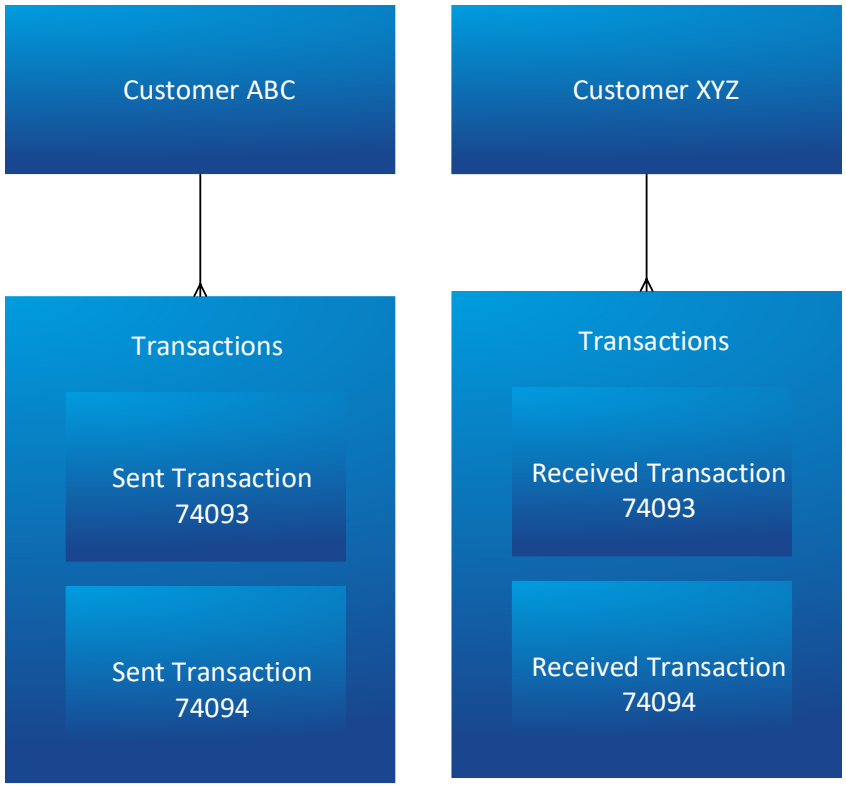
"The purpose of the relational model is to provide a declarative method for specifying data and queries: users directly state what information the database contains and what information they want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for answering queries."

Key formal modelling concepts: normal forms (e.g. 3NF), data integrity (PK constraints, FK constraints…)

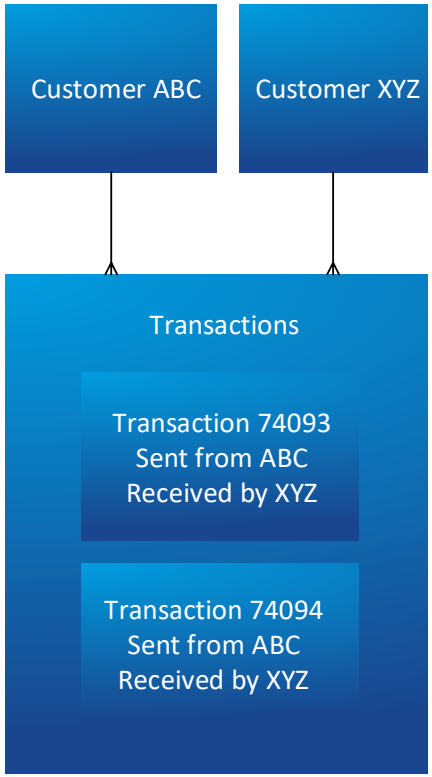
Source: https://en.wikipedia.org/wiki/Relational_model

Payments example – Hierarchical vs. Relational

Hierarchical



Relational



Pain potential after blunder:

SEVERE

Data Integrity

Data Integrity

ACID

- Atomicity – All-or-nothing
- Consistency – integrity (primary key constraints, foreign keys constraints)
- Isolation – concurrency control
  - Good intro - https://vladmihalcea.com/a-beginners-guide-to-read-and-write-skew-phenomena/
  - Deeper dive - https://medium.com/@andrew.gregovic/think-twice-before-dropping-acid-and-throw-your-cap-away-dbe0d6171dc0
- Durability – NOT losing data

Pain potential after blunder:

SEVERE

Big O, indexes, joins, sorts & query optimisation

Computational Complexity (a.k.a BigO)

Bubble Sort

```
public class BubbleSortExample {
    static void bubbleSort(int[] arr) {
        int n = arr.length;
        int temp = 0;
         for(int i=0; i < n; i++){
                 for(int j=1; j < (n-i); j++){
                          if(arr[j-1] > arr[j]){
                                  temp = arr[j-1];
                                  arr[j-1] = arr[j];
                                  arr[j] = temp;
                          }
                 }
         }
```

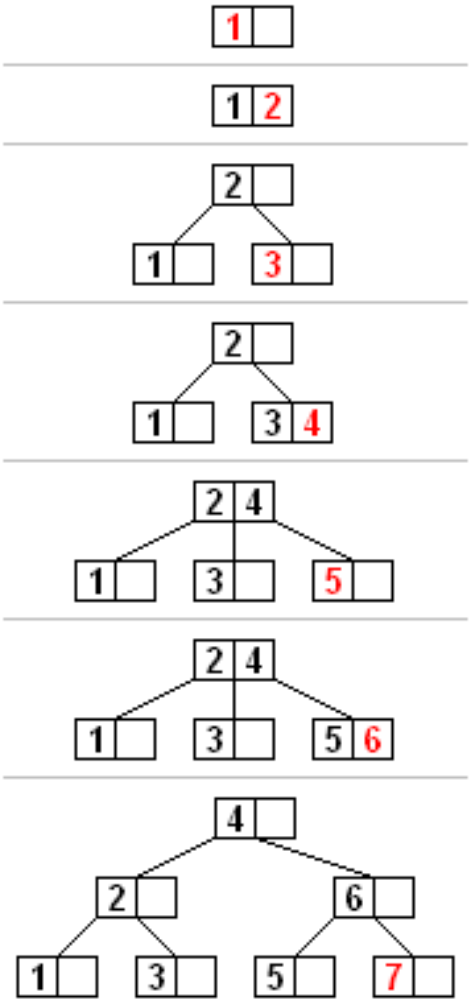
Bubble Sort = O (n^2)

Computational Complexity = typically f(x) of time output for a given x input.

Indexes

- Most common index: B-Tree (Online Transaction Processing systems)
- Variations: Index-organized tables, clustered (SQL Server), reverse key, Compressed Prefix B+-Tree
- Bitmap (data warehouses)
- Columnstore
- Text
- R-tree (spatial)
- Hash

Source: https://en.wikipedia.org/wiki/B-tree



Joins & Sorts

Physical Joins

- Nested loop O(n logn)
- Hash O(n)
- Merge O (n logn)

Sorts

- Merge O (n logn)
- Quicksort O (n logn)

Query optimisation

- A logical SQL query needs to be translated into physical execution. Optimisation = choosing the “best” execution plan
- All modern optimisers are **cost based**. Cost = mix of
  - time to execute
  - CPU time
  - I/O
- Heavily relies on data statistics (assumptions for the query). Assumptions can go wrong!
- Plenty of modern improvements (dynamic query plans, automatic query cancellations and retries)

Further reading

- https://en.wikipedia.org/wiki/Query_optimization

Pain potential after blunder:

MODERATE

SQL – Structured Query Language

SQL – Thinking in Sets

DDL – Data definition

- Schema maintenance (tables)
- Views / materialized views
- Indexes
- Users & permissions

DML – Data manipulation

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE a.k.a. UPSERT

Various editions of SQL standards, various level of support across different DBMSs!

Pain potential after blunder:

SEVERE

Goodies you'll learn @ work

Backup & Restoration

- Simple in theory, but in practice…
- Typical old-school approach: restore as of a timestamp, roll forward change logs
- Alternative approach: roll backward
- 2018+ approach: automate everything!
- 3 critical factors to successful backups & restorations:
  1. TEST
  2. TEST
  3. TEST

Pain potential after blunder:

CATASTROPHIC

Object-relational mappers

- Map application (e.g. Java/.Net/JS) POJOs to relational entities
- Help with simple cases
- Can be increasingly painful with more complex cases

Further reading

- https://en.wikipedia.org/wiki/Object-relational_mapping

Pain potential after blunder:

SEVERE

Not-so-goodies you'll learn @ work

NoSQL

Type	Examples
Key-Value Cache	Apache Ignite, Coherence, eXtreme Scale, Hazelcast, Infinispan, JBoss Cache, Memcached, Repcached, Velocity
Key-Value Store	ArangoDB, Flare, Keyspace, RAMCloud, SchemaFree, Aerospike, quasardb
Key-Value Store (Eventually-Consistent)	DovetailDB, Oracle NoSQL Database, Dynamo, Riak, Dynomite, Voldemort, SubRecord
Key-Value Store (Ordered)	Actord, FoundationDB, InfinityDB, Lightcloud, LMDB, Luxio, MemcacheDB, NMDB, TokyoTyrant
Data-Structures Server	Redis
Tuple Store	Apache River, Coord, GigaSpaces
Object Database	DB4O, Objectivity/DB, Perst, Shoal, ZopeDB
Document Store	ArangoDB, BaseX, Clusterpoint, Couchbase, CouchDB, DocumentDB, IBM Domino, MarkLogic, MongoDB, Qizx, RethinkDB
Wide Column Store	Amazon DynamoDB, Bigtable, Cassandra, Druid, HBase, Hypertable, KAI, KDI, OpenNeptune, Qbase

Schema-free databases?

No, there's no such a thing!