01- Database Introduction & Architecture

School of Computer Science
University of Windsor

Agenda

- Course Introduction
- Team Introduction
- Lecture
- Project hand-out
- Introductory lab on MongoDB

Submission Deadlines

- Forming Groups: **Jan. 15th, 2023 [11:59 PM]**

Introductory Questions

What is a database

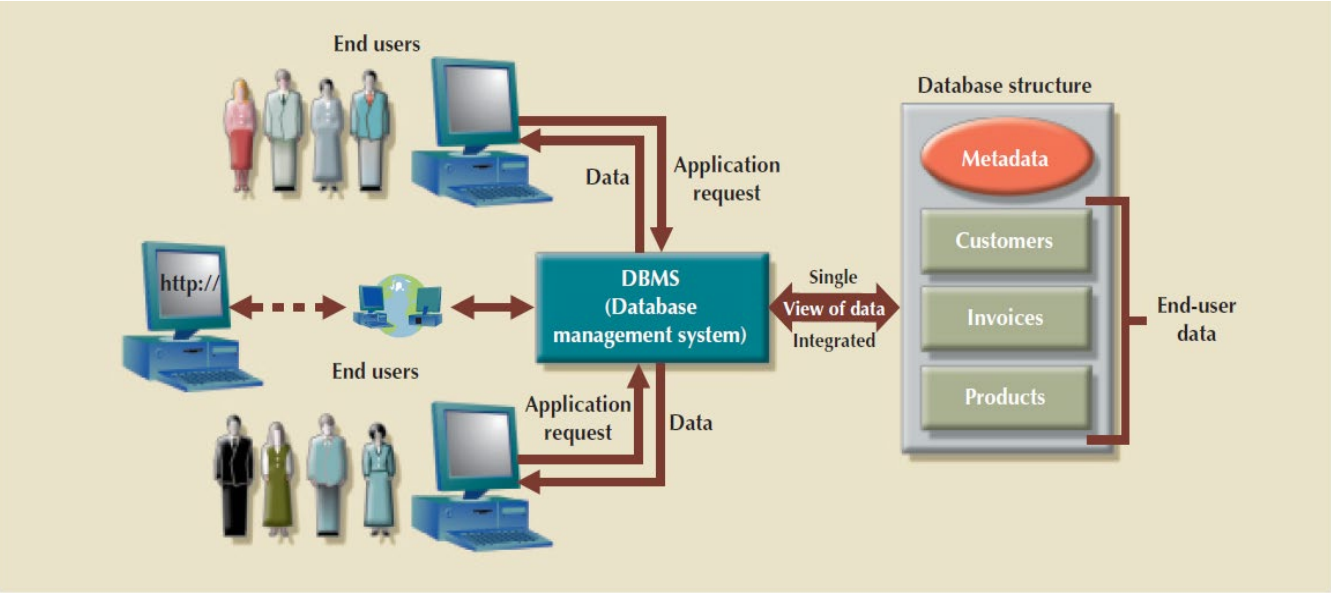
What is the history of databases?

What are the different database system architectures?

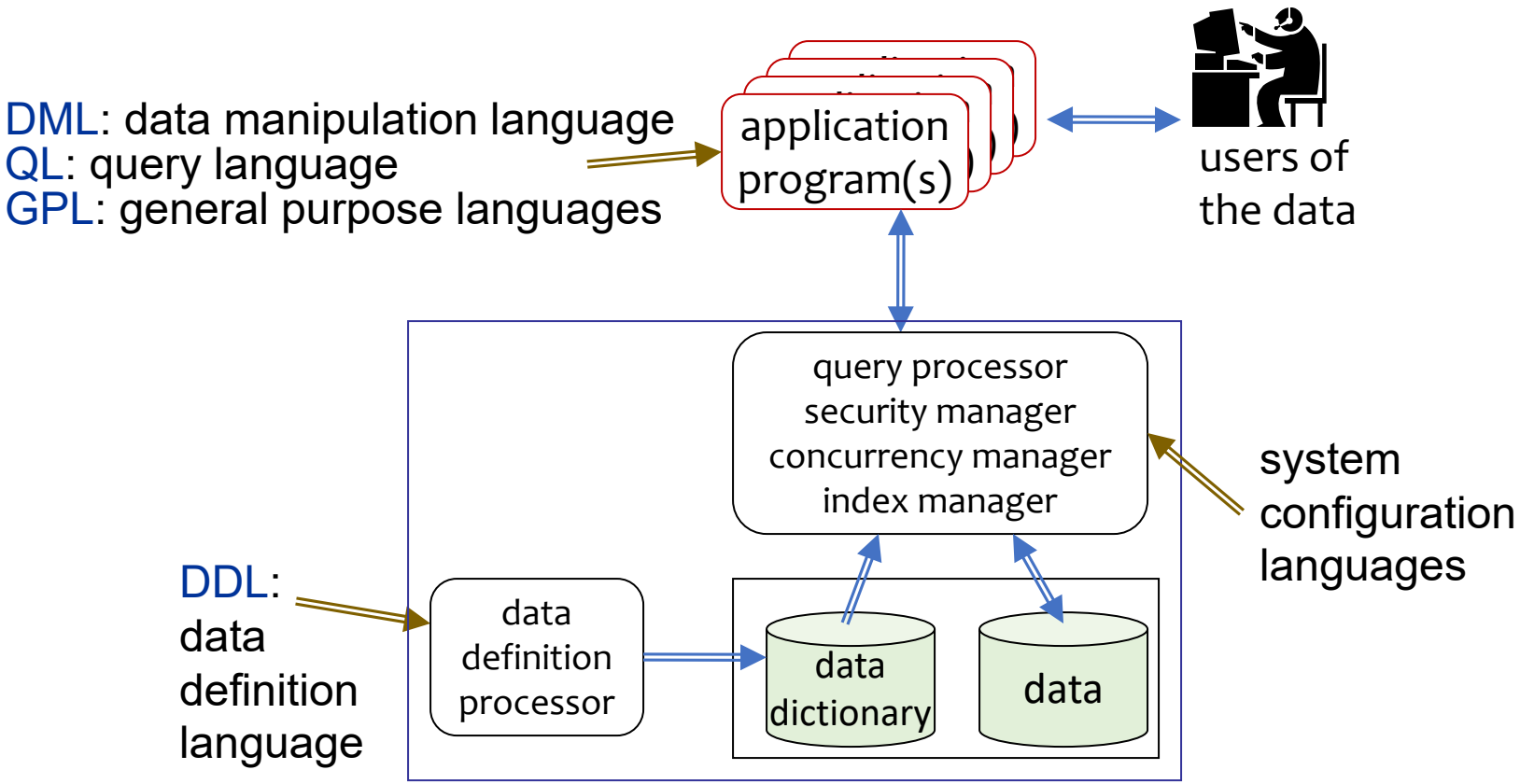
What is a Database?

- A database is any collection of data.

- A DBMS is a software that manages and controls access to the database.



DBMS Languages

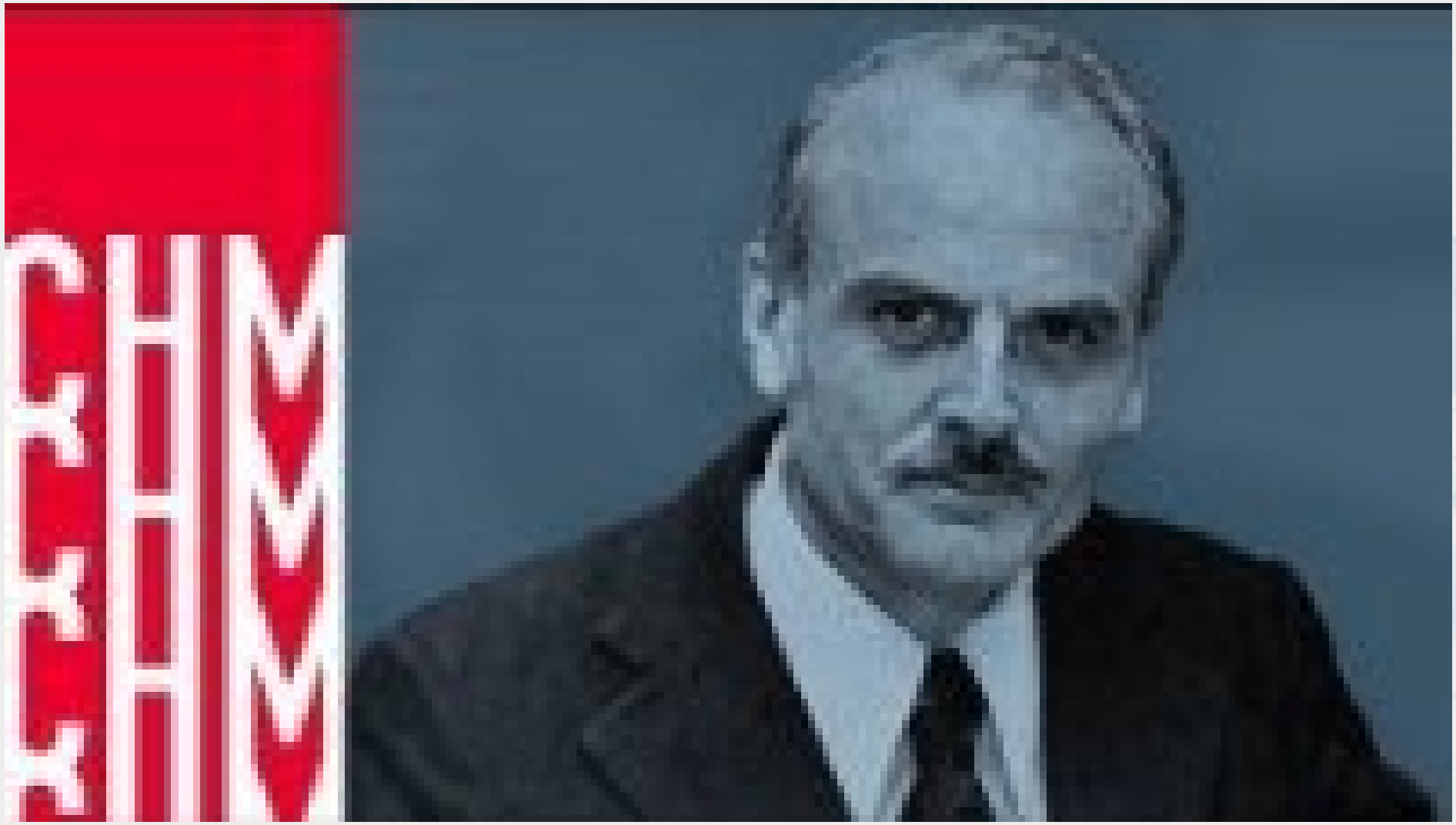


Advantages of Using a DBMS

Anything you can do with a DBMS, you can do with a file system, a network and a heap of C code. So why spend the money to buy a DBMS?

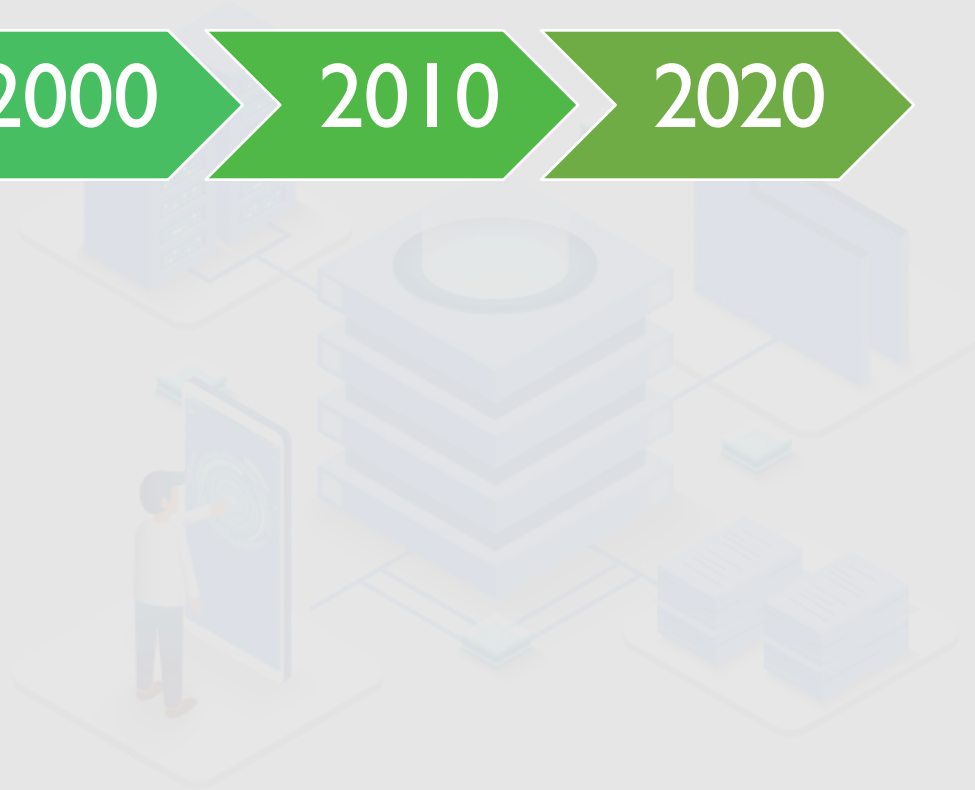
- **Integrity**: A DBMS maintains the consistency of stored data
- **Concurrency**: A DBMS supports access by concurrent users
- **Access Control**: A DBMS can restrict access to authorized users
- **Redundancy Control**: A DBMS can assist in controlling redundancy
- **Backup and Recovery**: A DBMS can provide backup and recovery.
  - *backup* = snapshots of the data particular times
  - *recovery* = restoring the data to a consistent state after a system crash

History of Database – Video



History of Databases - YouTube

1960 1970 1980 1990 2000 2010 2020

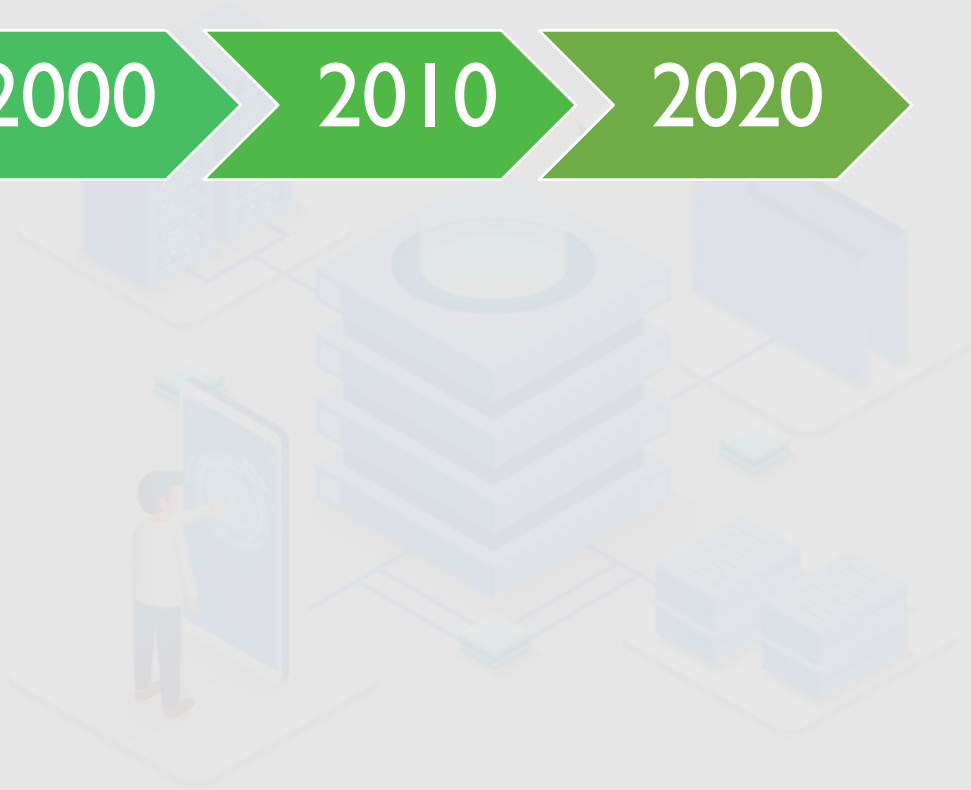


IDS

Honeywell - The first network DBMS.

Charles Bachman

1960 1970 1980 1990 2000 2010 2020



IDS

Honeywell - The first network DBMS.

Charles Bachman

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

idms

IDS

Honeywell - The first network DBMS.

Charles Bachman

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

IDS

Honeywell - The first network DBMS.

Charles Bachman

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** to store database in simple data structures, access data through high level language

Information Retrieval P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity
CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. Data Dependencies in Present Systems

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing depend-

IDS

Honeywell - The first network DBMS.

Charles Bachman

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** to store database in simple data structures, access data through high level language

System R, **INGRES, Oracle**

ORACLE

Informix TERADATA

IDS

Honeywell - The first network DBMS.

Charles Bachman

DB2

IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

1960 1970 1980 1990 2000 2010 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

System R, INGRES, Oracle

IDS

Honeywell - The first network DBMS.
Charles Bachman

DB2

IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

1960 1970 1980 1990 2000 2010 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

System R, INGRES, Oracle

IDS

Honeywell - The first network DBMS.
Charles Bachman

OBJECT-ORIENTED DATABASES

In 1983
DB2
IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

VERSANT

MarkLogic™

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

System R, INGRES, Oracle

IDS

Honeywell - The first network DBMS.

Charles Bachman

OBJECT-ORIENTED DATABASES

In 1983

DB2

IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

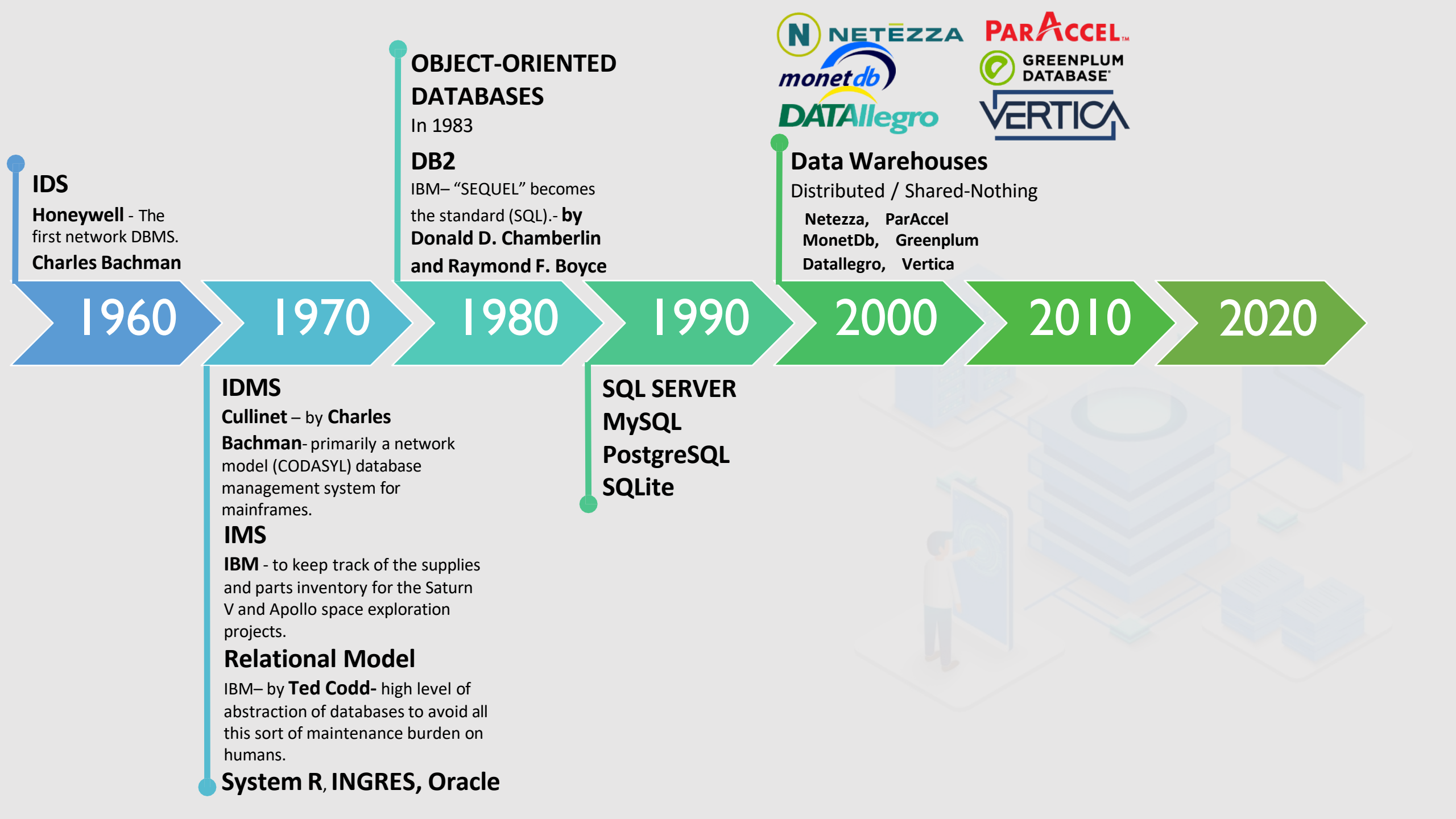
System R, **INGRES, Oracle**

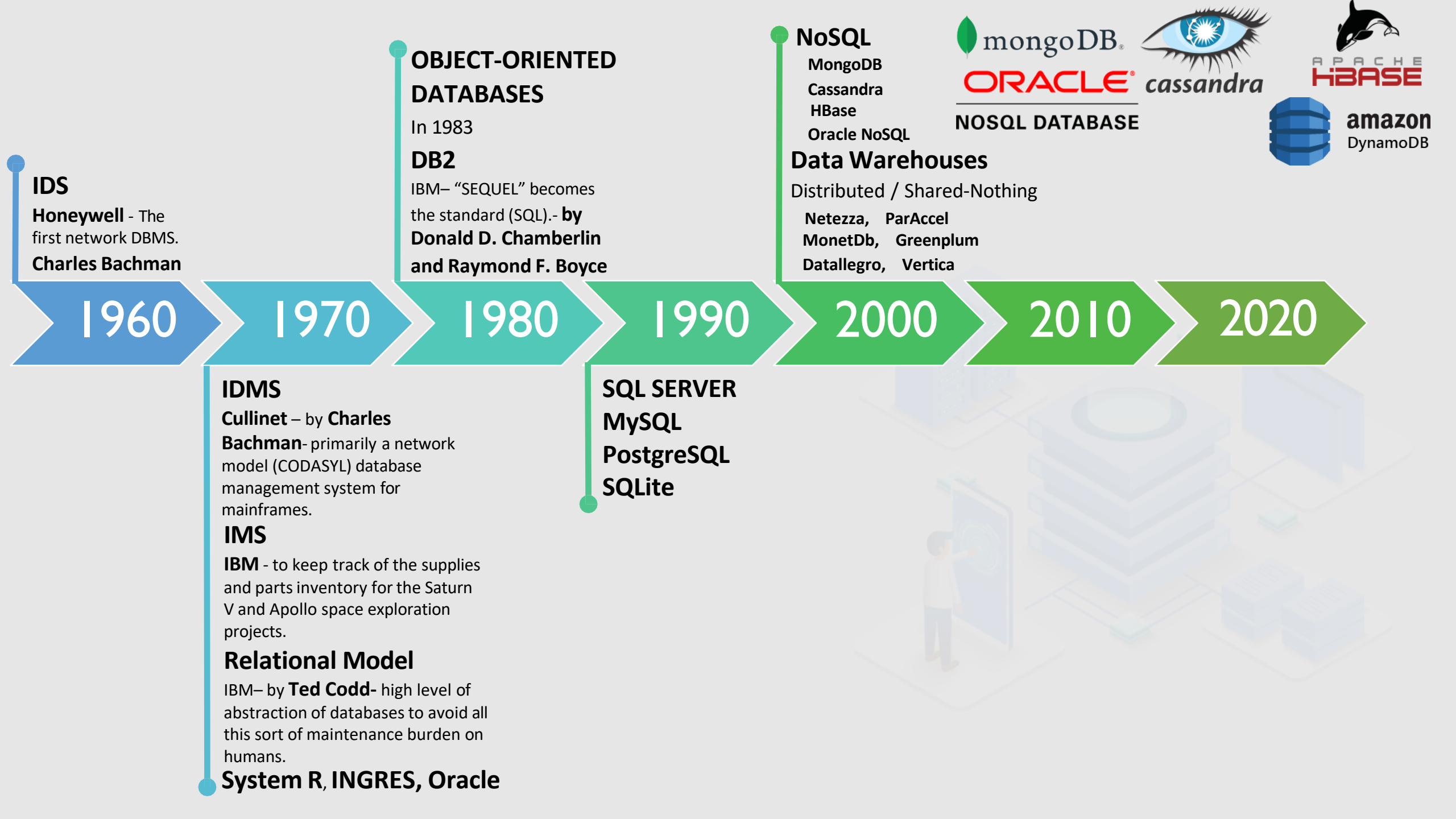
SQL SERVER

MySQL

PostgreSQL

SQLite


IDS
Honeywell - The first network DBMS.
Charles Bachman
1960
IDMS
Cullinet – by Charles Bachman- primarily a network model (CODASYL) database management system for mainframes.
IMS
IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.
Relational Model
IBM– by Ted Codd- high level of abstraction of databases to avoid all this sort of maintenance burden on humans.
System R, INGRES, Oracle
1970
OBJECT-ORIENTED DATABASES
In 1983
DB2
IBM– "SEQUEL" becomes the standard (SQL).- by Donald D. Chamberlin and Raymond F. Boyce
1980
SQL SERVER
MySQL
PostgreSQL
SQLite
1990
NETEZZA
ParAccel
monetdb
GREENPLUM DATABASE
DATAllegro
VERTICA
Data Warehouses
Distributed / Shared-Nothing
Netezza, ParAccel
MonetDb, Greenplum
Datallegro, Vertica
2000
2010
2020


IDS
Honeywell - The first network DBMS.
Charles Bachman
1960
IDMS
Cullinet – by Charles Bachman- primarily a network model (CODASYL) database management system for mainframes.
IMS
IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.
Relational Model
IBM– by Ted Codd- high level of abstraction of databases to avoid all this sort of maintenance burden on humans.
System R, INGRES, Oracle
1970
OBJECT-ORIENTED DATABASES
In 1983
DB2
IBM– "SEQUEL" becomes the standard (SQL).- by Donald D. Chamberlin and Raymond F. Boyce
1980
SQL SERVER
MySQL
PostgreSQL
SQLite
1990
NoSQL
MongoDB
Cassandra
HBase
Oracle NoSQL
Data Warehouses
Distributed / Shared-Nothing
Netezza, ParAccel
MonetDb, Greenplum
Datallegro, Vertica
2000
2010
2020
mongoDB.
ORACLE
NOSQL DATABASE
cassandra
APACHE HBASE
amazon DynamoDB

1960

IDS

Honeywell - The first network DBMS.

Charles Bachman

1970

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

System R, **INGRES, Oracle**

1980

OBJECT-ORIENTED DATABASES

In 1983

DB2

IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

1990

SQL SERVER

MySQL

PostgreSQL

SQLite

2000

NoSQL

MongoDB
Cassandra
HBase
Oracle NoSQL

Data Warehouses

Distributed / Shared-Nothing

Netezza, ParAccel
MonetDb, Greenplum
Datallegro, Vertica

2010

NewSQL

a relational database with the scalable properties of NoSQL

VoltDB, Clustrix, MemSQL, Spanner

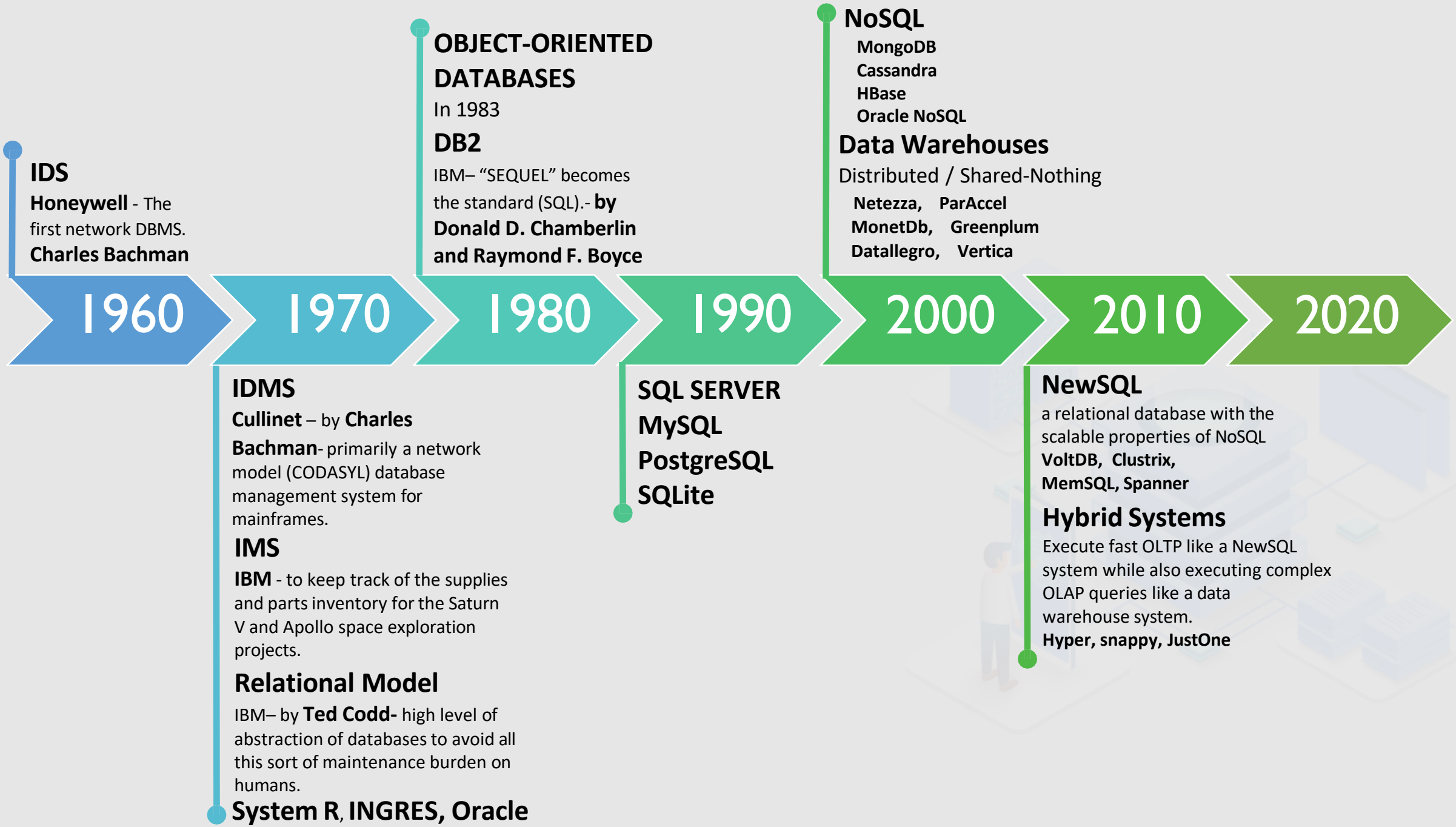
2020

VOLTDB

Clustrix

memSQL

Cloud Spanner


IDS
Honeywell - The first network DBMS.
Charles Bachman
1960
1970
IDMS
Cullinet – by Charles Bachman- primarily a network model (CODASYL) database management system for mainframes.
IMS
IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.
Relational Model
IBM– by Ted Codd- high level of abstraction of databases to avoid all this sort of maintenance burden on humans.
System R, INGRES, Oracle
1980
OBJECT-ORIENTED DATABASES
In 1983
DB2
IBM– "SEQUEL" becomes the standard (SQL).- by Donald D. Chamberlin and Raymond F. Boyce
1990
SQL SERVER
MySQL
PostgreSQL
SQLite
2000
NoSQL
MongoDB
Cassandra
HBase
Oracle NoSQL
Data Warehouses
Distributed / Shared-Nothing
Netezza, ParAccel
MonetDb, Greenplum
Datallegro, Vertica
2010
NewSQL
a relational database with the scalable properties of NoSQL
VoltDB, Clustrix, MemSQL, Spanner
Hybrid Systems
Execute fast OLTP like a NewSQL system while also executing complex OLAP queries like a data warehouse system.
Hyper, snappy, JustOne
2020

1960

IDS

Honeywell - The first network DBMS.

Charles Bachman

1970

IDMS

Cullinet – by **Charles Bachman**- primarily a network model (CODASYL) database management system for mainframes.

IMS

IBM - to keep track of the supplies and parts inventory for the Saturn V and Apollo space exploration projects.

Relational Model

IBM– by **Ted Codd-** high level of abstraction of databases to avoid all this sort of maintenance burden on humans.

System R, INGRES, Oracle

1980

OBJECT-ORIENTED DATABASES

In 1983

DB2

IBM– "SEQUEL" becomes the standard (SQL).- **by Donald D. Chamberlin and Raymond F. Boyce**

1990

SQL SERVER

MySQL

PostgreSQL

SQLite

2000

NoSQL

MongoDB
Cassandra
HBase
Oracle NoSQL

Data Warehouses

Distributed / Shared-Nothing

Netezza, ParAccel
MonetDb, Greenplum
Datallegro, Vertica

2010

NewSQL

a relational database with the scalable properties of NoSQL

VoltDB, Clustrix, MemSQL, Spanner

Hybrid Systems

Execute fast OLTP like a NewSQL system while also executing complex OLAP queries like a data warehouse system.

Hyper, nappy, JustOne

Cloud System

First database-as-a-service (DBaaS) offerings were "containerized" versions of existing DBMSs.

RedShift &Aurora from Amazon, Snowflake, Xeround

2020

amazon REDSHIFT

Amazon Aurora

snowflake

Rank			DBMS	Database Model
May 2022	Apr 2022	May 2021		
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document, Multi-model
6.	6.	↑ 7.	Redis	Key-value, Multi-model
7.	↑ 8.	↓ 6.	IBM Db2	Relational, Multi-model
8.	↓ 7.	8.	Elasticsearch	Search engine, Multi-model
9.	9.	↑ 10.	Microsoft Access	Relational
10.	10.	↓ 9.	SQLite	Relational
11.	11.	11.	Cassandra	Wide column
12.	12.	12.	MariaDB	Relational, Multi-model
13.	13.	13.	Splunk	Search engine
14.	14.	↑ 27.	Snowflake	Relational
15.	15.	15.	Microsoft Azure SQL Database	Relational, Multi-model
16.	16.	16.	Amazon DynamoDB	Multi-model
17.	17.	↓ 14.	Hive	Relational

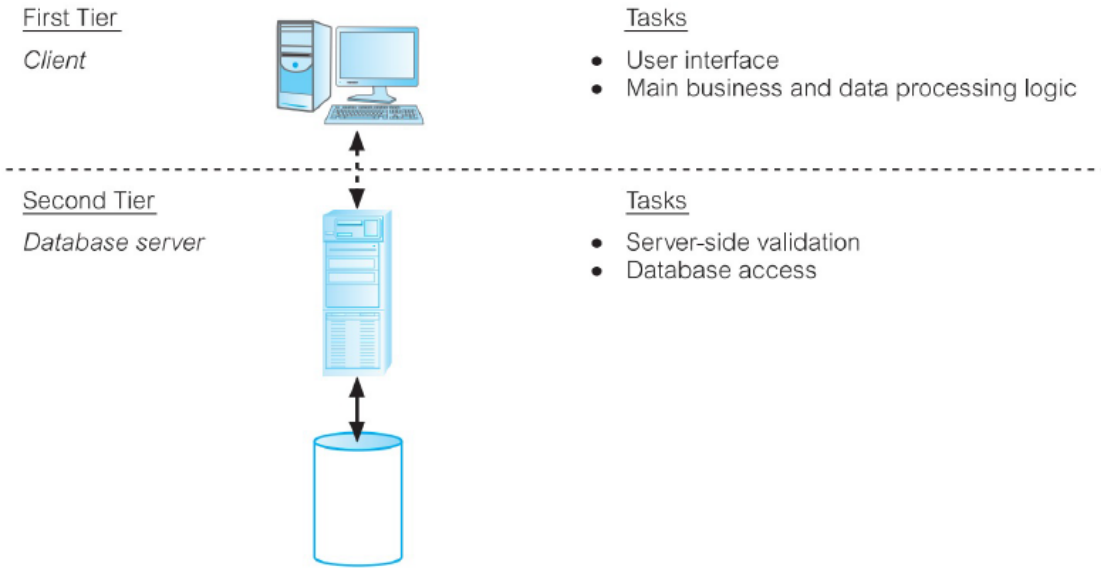
https://db-engines.com/en/ranking

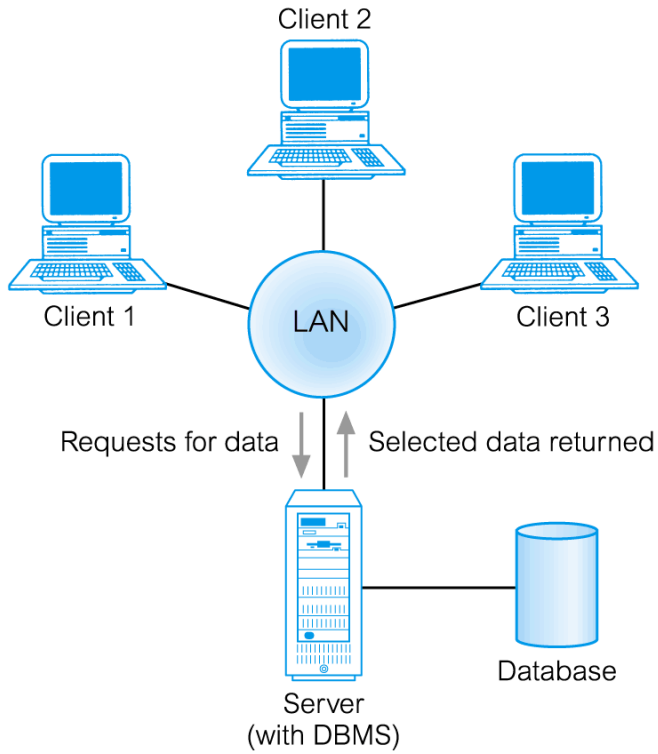
Multi-user DBMS Architectures

- **Two-Tier Client-Server**

  Client (tier 1) manages user interface and runs applications.

  Server (tier 2) holds database and DBMS.



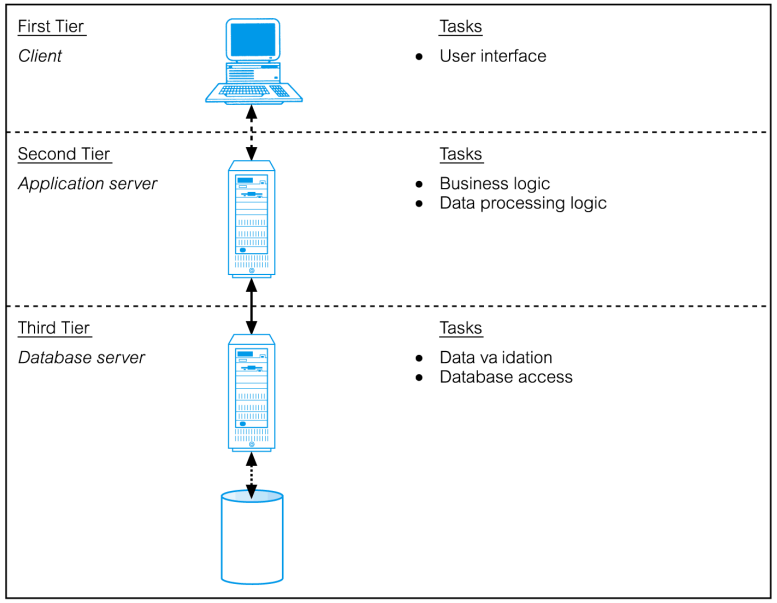


Multi-user DBMS Architectures

- **Three-Tier Client-Server**

  The need for enterprise scalability challenged the traditional two-tier client–server model.

- Client side presented two problems preventing true scalability:
  - ‘Fat’ client, requiring considerable resources on client’s computer to run effectively.
  - Significant client-side administration overhead.



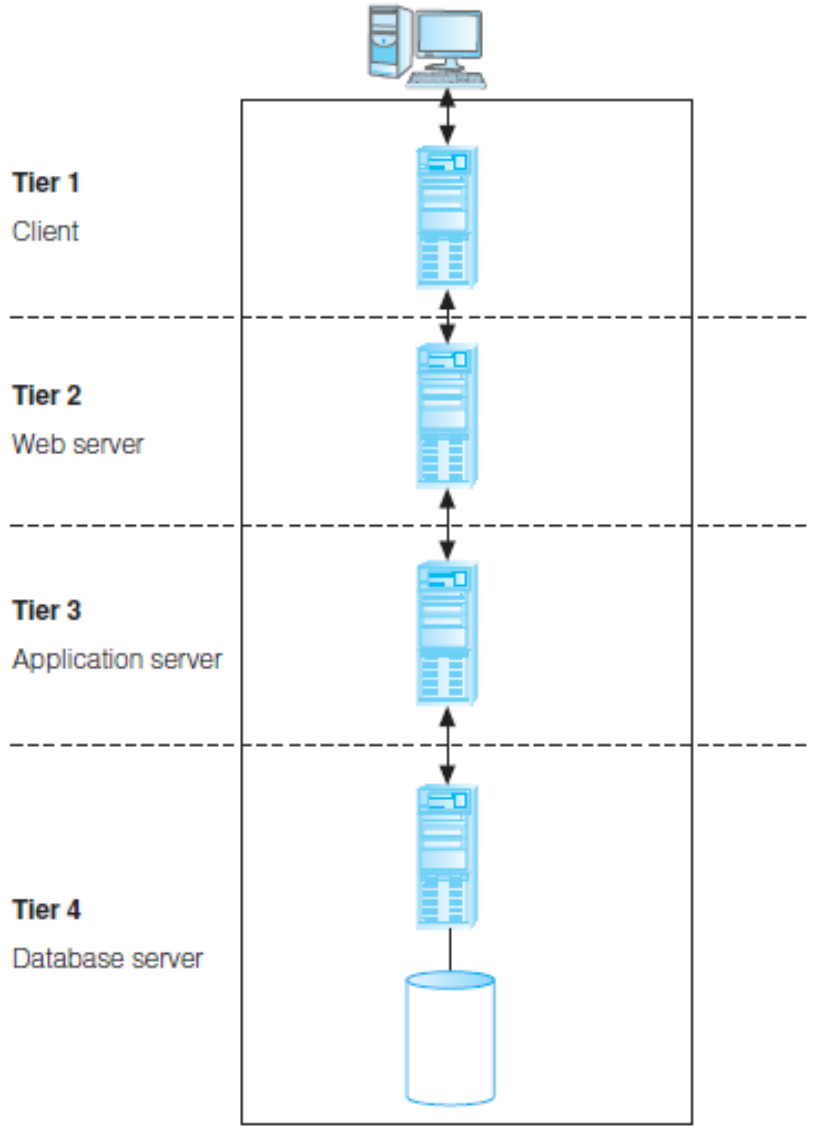
Source: Pearson Education

Multi-user DBMS Architectures

- **n-Tier Client-Server (e.g. 4-Tier)**

  The three-tier architecture can be expanded to *n* tiers, with additional tiers providing more flexibility and scalability.

  - Applications servers host API to expose business logic and business processes for use by other applications.



Source: Pearson Education

Distributed DBMSs

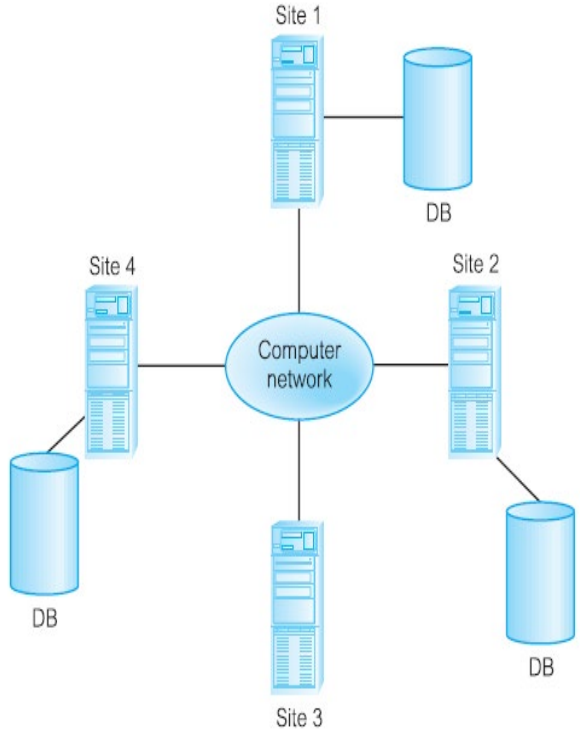
- A distributed database is a logically interrelated collection of shared data (and a description of this data), physically distributed over a computer network.

- A distributed DBMS is the software system that permits the management of the distributed database and makes the distribution transparent to users.

  - A DDBMS consists of a single logical database split into a number *of fragments*.

  - Each fragment is stored on one or more computers *(replicas)* under the control of a separate DBMS, with the computers connected by a network.

  - Each site is capable of independently processing user requests that require access to local data (that is, each site has some degree of local autonomy) and is also capable of processing data stored on other computers in the network.

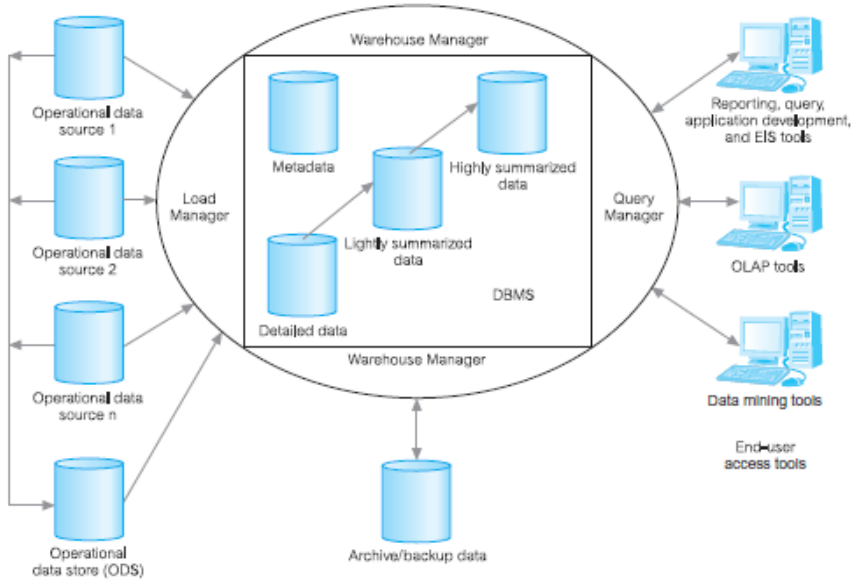


Source: Pearson Education

Data Warehousing

- A consolidated/integrated view of corporate data drawn from disparate operational data sources and a range of end-user access tools capable of supporting simple to highly complex queries to support decision making.

- Data Warehouse Video



Source: Pearson Education

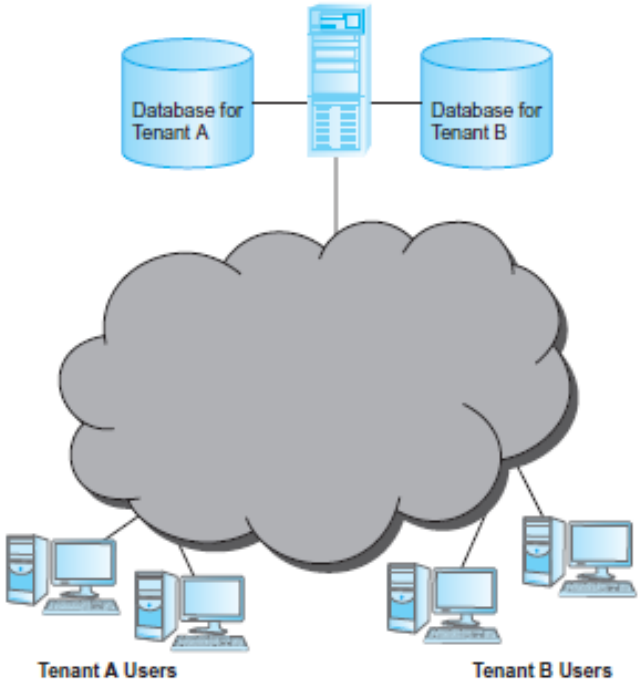
University of Windsor

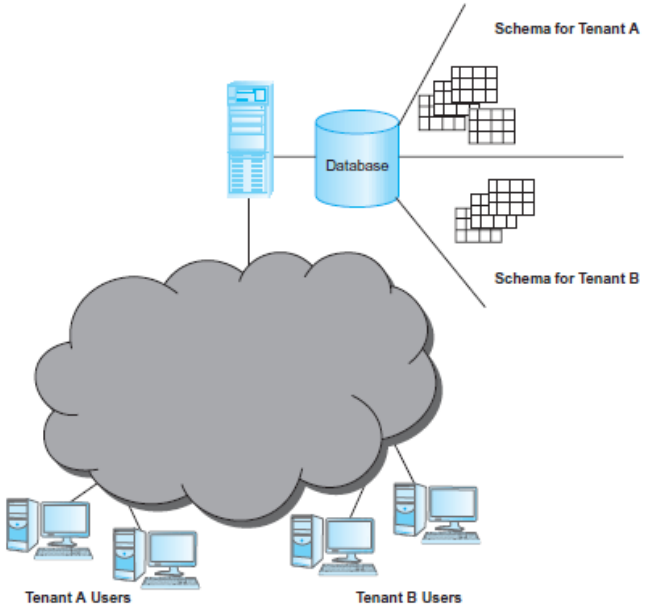
Cloud-based database solutions

- DBaaS offers full database functionality to application developers.
- A management layer is responsible for the continuous monitoring and configuring of the database.
- Helps to achieve optimized scaling, high availability, multi-tenancy (that is, serving multiple client organizations), and effective resource allocation in the cloud.
- Spares the developer from ongoing database administration tasks.

Cloud-based database solutions

- Multi-tenant cloud database
  - shared DBMS server, separate databases.
  - shared database, separate schema architecture





Source: Pearson Education

Recap and Conclusion

- Introduction
- Database Structure
- Advantages of the Database Approach
- History of Database Systems
- Multi-User DBMS Architecture
- Distributed DBMS
- Data Warehousing
- Cloud-based Database Solutions

Test your understanding

- Which of the following is **not** a valid NoSQL database?
  - A. Cassandra
  - B. HBase
  - C. MongoDB
  - D. PostgreSQL

Test your understanding

A benefit of the three-tier architecture is which of the following?

A. Results in a thinner client and database server
B. Performance improves for compiled SQL statements
C. New modules can be built to support specific business needs
D. All of the above.

Test your understanding

Which of these statement is **not** true about NewSql?

A. It introduces new implementation to traditional relational databases.

B. It brings together the advantages of SQL and NoSQL.

C. It is easy to migrate between the type and needs of the user.

D. These systems are better optimized for non-relational data.

Project Intro

Any Questions