L01: Introduction to Database Systems

DSAN-6300 & PPOL-6810

Databases & SQL Programming

Irina Vayndiner

August 28 & 31, 2023

GEORGETOWN UNIVERSITY

Outline for today's class

Welcome to DSAN-6300 / PPOL-6810

- Introductions
- Course Overview
- Introduction to Database Systems
  - What is a Database System
  - Application and Purpose of Database Systems
  - History of Database Systems
  - Data Models
  - Database Languages
  - Database Architecture

Introducing your teacher

Irina Vayndiner
MITRE

Email: iv95@georgetown.edu

Interests:
- Databases, Big Data
- Storytelling
- Improvisational theater (long-form)

Other courses I taught/teach at Georgetown:
- Massive Data Fundamentals (ANLY-502, HIDS-511, PPOL-567)
- Storytelling with Data (ANLY-599)

Introducing TAs

TAs

- **Raghav Sharma**
- **Ramdayal Rewaria (Lead DSAN)**
- **Yanyan Li**
- **Parsa Keyvani**
- **Shenghao Wang**
- **Pablo Jimenez**
- **Peijin Li (Lead PPOL)**

TA Office Hours & locations: published on Canvas

Students Introductions

STUDENTS POLL

Overall Course Topics

- Introduction to Relational Models and Schema Design
- SQL Programming (from beginners to advanced!)
- How databases work
  - Query Processing & Optimization methods
  - Concurrency and Transactions
  - Database Architectures
- Data storage including column-oriented and distributed storage
- Intro to SQL Big Data Concepts and Processing
- NoSQL and other non-relational databases

Course Tools

- Hands-on software and tools
  - MySQL Workbench on your laptop
  - RDS in AWS Cloud
  - MongoDB in AWS Cloud

Class Logistics

- **Classroom**
  - Mon: CBN #204
  - Tue: St Marys #110
- **Lectures & Labs Day/Time**
  - **Mon** for ANLY-6300-01: 3:30-6pm EST
  - **Thu** for ANLY-6300-02 and PPOL-6810: 12:30-3pm EST
- There will be some exceptions
  - Check Canvas announcements regularly!

Class Logistics (continued)

- Typically, each class will involve:
  - Lecture
  - Break
  - Lab hands-on work (based on the Lecture material)
- Communication outside of class:
  - Use Discussion boards on Canvas if ran into an issue or have a question
  - Use email for administrative issues
    - Need to inform
      - <u>NLT 10am the day of the class</u> if not able to attend class or attending on Zoom
      - NLT <u>3pm</u> on the day assignment is due
    - When you email me, please cc at least one TA
      - Ram for DSAN, Peijin for PPOL
      - Unless there is an issue that is not appropriate for TAs
- Please plan to use your laptops for every class

Class Materials on Canvas

We will use Canvas for the Class materials

- Announcements
    - Check frequently, or sign up for email alerts
- Syllabus
- Modules, including dates of tests
    - Data of the last test might shift!
- Lecture Slides (published after the lecture!)
- Assignments, e.g.
    - Usually will publish a reminder when an Assignment is published
- Deadlines
- Grades

Important: all Sections of this Course on Canvas will look like one section: ANLY-6300-01: You are still in the section you originally signed for!

Class Policies

- Class participation in classroom is expected, it is part of the grade
  - Ask & answer questions
- Assignments are due **on**
  - **HWs & Quizzes: Tuesdays at 11:59pm**
  - Usually around 10-14 days for HWs, one week for Quizzes
- Labs are due after the class on
  - **Thursday 11:59pm for Mon class**
  - **Tuesday 11:59pm for Thu class**
- **No late** submission for **Quizzes or Project**
- Canvas Locks at the due time!
- All assignments are <u>individual</u> assignments
- Follow the Georgetown Student Pledge
  - Will be painful if you do not!

Class Late Submission Policy

- Late Submission Policy
  - All non-test deliverables can be up to **24 hours late with NO PENALTY**.
  - Any assignment submitted after that grace period will result in **reduction of points** as follows:
    - HW Assignments: -5% per day for up to 5 days
    - All other Assignments: -10% per day for up to 5 days
  - The Midterm and Final text **CANNOT** be rescheduled
    - **Midterm: Thu 10/19 and Mon 10/23**
    - **Final test on zoom**: **Tue 12/5 at 10:30am for all 3 sections**
    - Please make a note of these dates NOW

DSAN-6300 / PPOL 6810, by the numbers

Years taught:	2019	2020	2021	2022	2023
Class sessions:	13	14	13	14	14
Class length:	2hrs 30min	2hrs 30min	2hr 30min	2h 30min	2h 30min
Enrolled students:	26	72	92	95	96
Num of Sections	1	2	2	3	3
Homework assignments:	3	3	3	3	3
Mini-Project:	1	1	1	1	1
Online Quizzes:	4	4	4	4	4
Labs	0	9	9	9	9
In-Class Tests	2 (Closed books)	2 (Open books)	2 (Open books)	2 (Closed & Open Books)	2 (Closed and Open books)

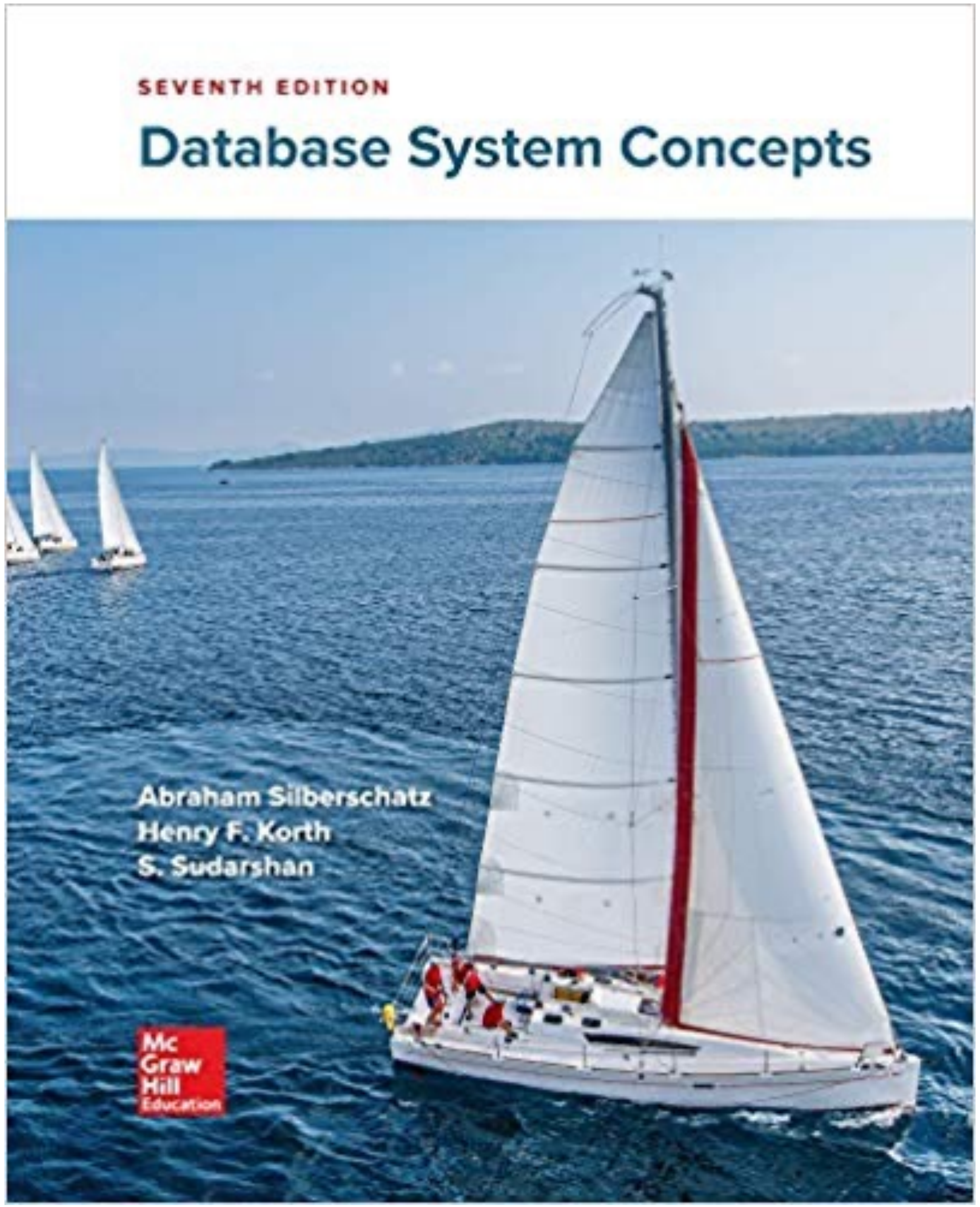
Tentative Class Dates, check Canvas for updates!

	DSAN-6300-01 Mon	**DSAN-6300-02& PPOL-6810 Thu**
1	8/28	8/31
2	9/5 **(Tue!)**	9/7
3	9/11	9/14
4	9/18	9/21
5	9/25 (recording)	9/28
6	10/2	10/5
7	10/16	10/12
8 (midterm)	**10/23**	**10/19**
9	10/30	10/26
10	11/6	11/2
11	11/13	11/9
12	11/20	11/16
13	11/27	11/30
14 (test)	**12/5 (Tue, 10:30am-1pm)**	**12/5 (Tue, 10:30-1pm)**

No classes
Mon, 9/4 Labor Day
Mon, 10/9: Mid Semester Holiday
Mon, 9/25: Yom Kippur

Class added: Mon 12/4 for in lieu of 8/24 (all 3 sections)

Textbook



- **7th Edition(2020)**
- 6th edition is outdated!
- Will use this book's materials for many of the Course lectures

Class Deliverables: What you need to do!

Course Activity	Weight
Class participation including in-class Labs	12%
Four Homeworks, including one Project	40%
Four On-line Quizzes	8%
Two In-Class Tests (midterm 15% and “final” 25%)	40%

Grading

- Final point count is 100%, the letter grade will follow standard guidelines for Fall 2023

Grades

- A: >= 92.5
- A-: 89.5 - 92.49
- B+: 87.99 - 89.49
- B: 81.5 - 87.98
- B-: 79.5 - 81.49

Grading Rubric for Homework Assignments & Project

- We will look at the results files and the scripts.
- If the result files are exactly what is expected, in the proper format, etc., we may run your scripts to make sure they produce the output. If everything works, you will get full credit for the problem.
- If the results files are not what is expected, we will look and run your code and provide partial credit wherever possible and applicable.
- Points **will** be deducted for each/any the following reasons:
  - Instructions are not followed (please read assignments carefully!)
  - Output is not in expected format (e.g. not sorted, missing fields, wrong delimiter, unusual characters in the files, etc.)
  - There are more files delivered than need to be
  - There are additional lines in the results files (whether empty or not)
  - Files are not the requested filename
  - The Assignment is late: see late submission policy

What you need to take this course

- Textbook
- Commitment to Assignments
  - Quiz will take up to 1hr
  - Finishing the Lab will take about 2hrs on avg
  - **HW will take more time**– plan ahead!
- Use of Amazon Educate Cloud (we will explain!)
  - We will use Amazon Web Services (AWS) for hands-on work
- Hardware:
  - Your laptop (OS: Mac or Windows w/ Cygwin)
    - If you have OS Windows on your laptop, please email TA Paolo
- Software (Installation Instructions will follow)
  - MySQL Workbench
  - RDS in AWS Cloud
  - MongoDB in AWS Cloud

Tentative Schedule for Fall 2023: Monitor Canvas for updates

Class num	Class	Labs (due **Tue** on the week after the class)	Book Chapters	Homeworks	Quizzes	Due on **Tue** of the week
1	L1. Introduction, Database Systems		1			
2	L2. Database Design Using the E-R Model	Lab: Setup	6		Q1	
3	L3: Relational Model and Relational Algebra	Lab: Setup	2, 7	HW1		Q1
4	L4 SQL 1	Lab: SQL 1	3			
5	L5. SQL 2	Lab: SQL 2	3, 4	HW2	Q2	HW1
6	L6. SQL 3	Lab: SQL 3	4,5			Q2
7	L7. Advanced Topics in SQL and Database Design	Lab: Adv Topics	15, 16	HW3		HW2
8	Midterm					H2
9	L9. Intro to Big Data	Lab: TBD	10, 11, 12, 14		Q3	HW3
10	L10. Non-relational DB + Guest Speaker	Lab: MongoDB		HW4 (Project)		
11	L11. Data Warehousing, Data Analytics, and Transactions	Lab: Analytics	TBD			Q4
12	L12. Database Optimization	Lab: Db Optimization	11, 15		Q4	Q3
13	L13. Data Management for the Enterprise					
14	Test					Project

Why you need to take this course

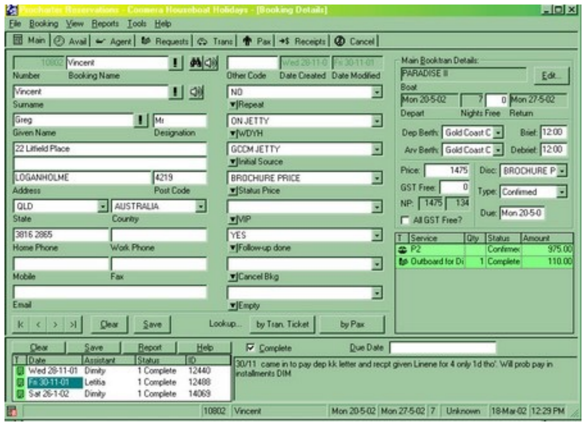
Languages
R, SAS, Python, Matlab, SQL,
Hive, Pig, Spark
Skills & Talents
✓ Distributed computing
✓ Predictive modeling
✓ Story-telling and visualizing
✓ Math, Stats, Machine Learning
DATA SCIENTIST
"AS RARE AS UNICORNS"
Role
Cleans, massages and organizes (big) data
Mindset
Curious data wizard
HIRED BY
Google
Microsoft
Adobe

"As I looked around at some open data science positions, I think about 99% of them required a working knowledge of SQL"

Introduction to Database Systems

Databases are everywhere

- A **database** is a collection of data, managed over a period of time
  - <u>Big</u>: Google web crawls, sales records at Walmart
  - <u>Small</u>: a personal Christmas card list, a recipe file
  - <u>Specialized</u>: medical images, Facebook links, a hybrid auto design
  - <u>You</u> are in Georgetown student database
- Databases are increasingly found in unexpected places
  - Are your photos in Apple cloud?
  - There is a personal autocomplete data on your phone or email
  - Sometimes people are not aware they use a database since accessing database forms is part of everyday life



Number	10802
Booking Name	Vincent
Other Code	
Date Created	Wed 28-11-0
Date Modified	Fri 30-11-01
Surname	Vincent
Given Name	Greg
Designation	Mr
Address	22 Litfield Place, LOGANHOLME
Post Code	4219
State	QLD
Country	AUSTRALIA
Home Phone	3816 2865
Work Phone	
Mobile	
Fax	
Email	
Repeat	NO
WDYH	ON JETTY
Initial Source	GCCM JETTY
Status Price	BROCHURE PRICE
VIP	
Follow-up done	YES
Cancel Bkg	
Empty	

Main Booktran Details:

Boat	PARADISE II
Depart	Mon 20-5-02
Nights	7
Free	0
Return	Mon 27-5-02
Dep Berth	Gold Coast C
Brief	12:00
Arv Berth	Gold Coast C
Debrief	12:00
Price	1475
Disc	BROCHURE P
GST Free	0
Type	Confirmed
NP	1475 134
Due	Mon 20-5-0
All GST Free?	☐

T	Service	Qty	Status	Amount
	P2		Confirmed	975.00
	Outboard for Di	1	Complete	110.00

T	Date	Assistant	Status	ID
	Wed 28-11-01	Dimity	1 Complete	12440
	Fri 30-11-01	Letitia	1 Complete	12488
	Sat 26-1-02	Dimity	1 Complete	14069

☑ Complete   Due Date

30/11 came in to pay dep kk letter and recpt given Linene for 4 only 1d tho'. Will prob pay in instalments DIM

10802 | Vincent | Mon 20-5-02 | Mon 27-5-02 | 7 | Unknown | 18-Mar-02 | 12:29 PM

Why are Databases Important to Study

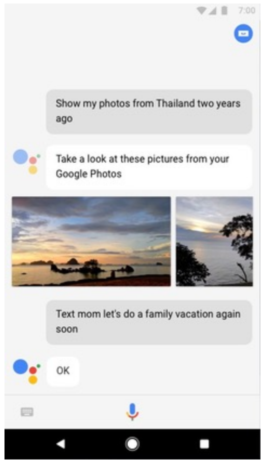
- Lots of benefits from understanding data principles
  - Databases are crucial components of complex software systems
    - How does Walmart decide what to put on shelves?
    - How Amazon decides what to recommend to you?
    - How does 7-11 decide what to put on the shelves next to each other?
  - Database perspective will allow you to understand better how the whole system works
  - The principles themselves may even be your biggest takeaway from this course…

Some Database Examples

- Databases touch all aspects of our lives
- Enterprise Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.
- Manufacturing: management of production, inventory, orders, supply chain.
- Banking and finance
  - Customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data
- Universities: registration, grades, etc.

Examples of Data in Everyday Life

- Facebook, LinkedIn, Tweeter, SnapChat, etc.
- Targeted advertising
- Google Assistant ("Hey Google"), Siri, Alexa
  - "Gets you just the right information at just the right time"
- Wearables:
  - Smart Glasses: wearable computer with a head-mounted display
    - Google Glass (one of the first):
      - Why didn't it succeed?
      - Resurfaced as Glass Enterprise Edition
    - Currently many types of Smartglasses
    - Apple Watch, etc.
    - Health monitoring wearables
- Netflix
  - Uses Data to predict if TV show will be a hit with audiences before it is produced
- Virtual reality
  - Data researchers are betting on VR technologies to explore new ways to visualize and analyze complex and dynamic datasets.


Show my photos from Thailand two years ago

Take a look at these pictures from your Google Photos

Text mom let's do a family vacation again soon

OK

Examples of Government Data

- Sensor Data
- Biosecurity
- Emergency preparedness
- Logistics, e.g. for optimized decisions about logistics
- Economic Data, e.g. unemployment rate
- Financial Data, e.g. compliance & fraud detection
- Climate modeling
- Environmental Data, e.g. environmental monitoring & mitigation
- Cybersecurity
- Healthcare, e.g. public health, epidemics early detection

- Big Data Research is supported by Government, for example:
  - Department of Energy (DoE)
  - National Science Foundation (NSF)

What is a DBMS

- We talked about databases. What is DBMS?
- A DBMS is a DataBase Management System
  - A software tool making it easier to manage databases
  - Over 50 years of research and engineering behind modern DBMSs
  - A huge business (billions of $ per year)
  - In this course we will cover RDBMS (Relational DBMS), and also other DBMS's (nosql, semi-structured, key-value, etc)

- Q: Can we have a database without a DBMS?

What is a DBMS (continued)

- A database system (DBMS) is a collection of interrelated data and a set of programs that allow users to access, manage and use these data.
- Major DBMS elements are:
  - Data models
    - A collection of tools for describing data
  - Data access
    - Structured Query Language (SQL)
    - Language, such as C/C++, Java or Python, with embedded SQL queries that access the data in the database
    - Application programs that are used to interact with the database in this fashion.
      - Important term: Application Programming Interface (API)

Why use DBMS

- Let's say you have 1TB of Public Policy data
- You could just write an application that stores data in disk files
- Some issues you would need to address:
  - Speed: linear access can be slow, optimization and indexing can provide dramatic speedup
  - Query/report interface: you need to write a program for every question; is everybody who needs information a developer?
  - Schema: how does a new user know where to find "Criminal Justice Policy"?
  - Data integrity: if you delete a topic, how do you ensure all related items are deleted?
  - Security: how do you limit who can query sensitive information?
  - Transactions: you may need to code up protection for concurrent users "stepping on" each other, and also against system crashes
- That can be **a lot of work!**

Challenges of Database Applications

In the early days, database applications were built directly on top of file systems, which led to

- Data redundancy and inconsistency
  - Data is stored in multiple file formats resulting in duplication of information in different files
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation
  - Consistency and completeness of data retrieved
- Integrity problems
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones

Challenges of Database Applications (continued)

- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Ex: Two people reading a balance (say $100) and updating it by withdrawing money (say $100 each) at the same time
- Security problems
  - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

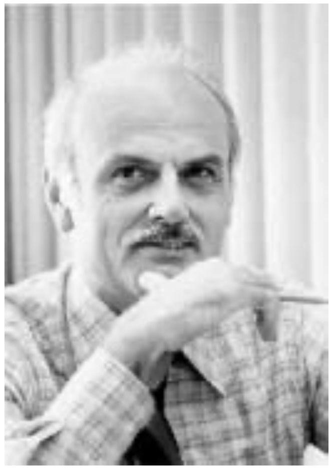
Services that DBMS Offers

- A Standard Query Language (SQL)
- Automated query optimization, and sub-linear access through indexing
- Schematic data modeling / design
- Automated data integrity
- Security (access controls, audit logs, encryption, authentication ...)
- Crash-proof Transactions
- Concurrency control
- Massively parallel optimizations
- GUI’s, APIs (e.g., JDBC), and a rich set of development tools
- Import/export: XML, JSON, CSV
- Sophisticated administration and tuning tools

If your application needs these, an DBMS is a powerful time-saving tool

History of Database Systems

- 1832 — Semen Korsakov uses punch cards for data storage in Russia
- 1890 — Herman Hollerith used paper punch cards for US 1890 Census
  - 1896 Hollerith founded Tabulating Machine Company, which later becomes IBM
- 1950s and early 1960s
  - Data processing using magnetic tapes for storage; punch cards for input
  - 1964 — SABRE system goes online (American Airlines & IBM).
    - Two database models: network model (CODASYL) and hierarchical model (IMS)
- Late 1960s and 1970s
  - 1970-72 Edgar Frank "Ted" Codd defines the Relational Data Model
    - Codd won the ACM Turing Award for this work
    - IBM Research begins System R prototype
  - UC Berkeley (Michael Stonebraker) begins Ingres prototype
  - Oracle releases first commercial relational database
  - High-performance (for the era) transaction processing



INGRES

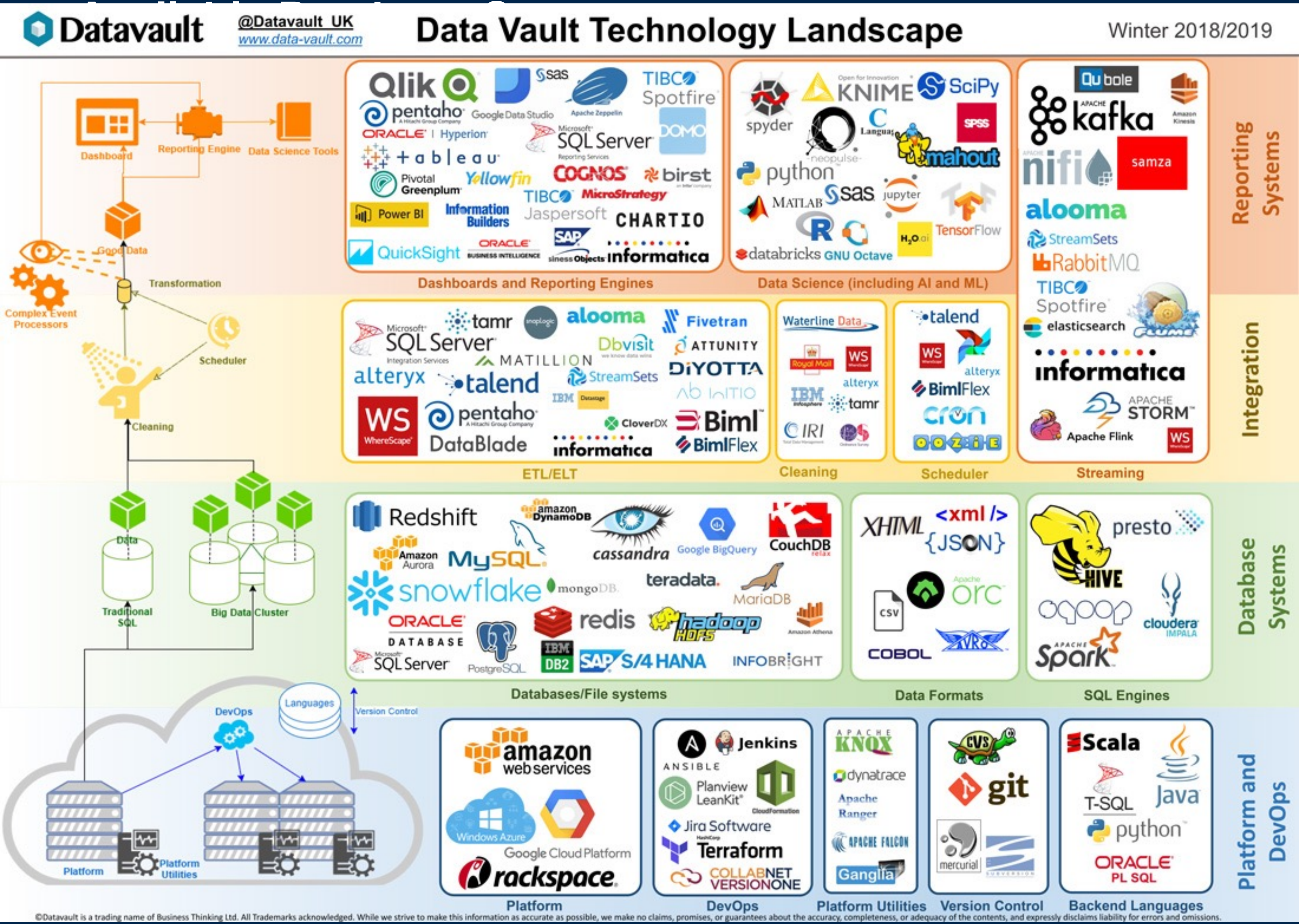
ORACLE

History of Database Systems (Continued)

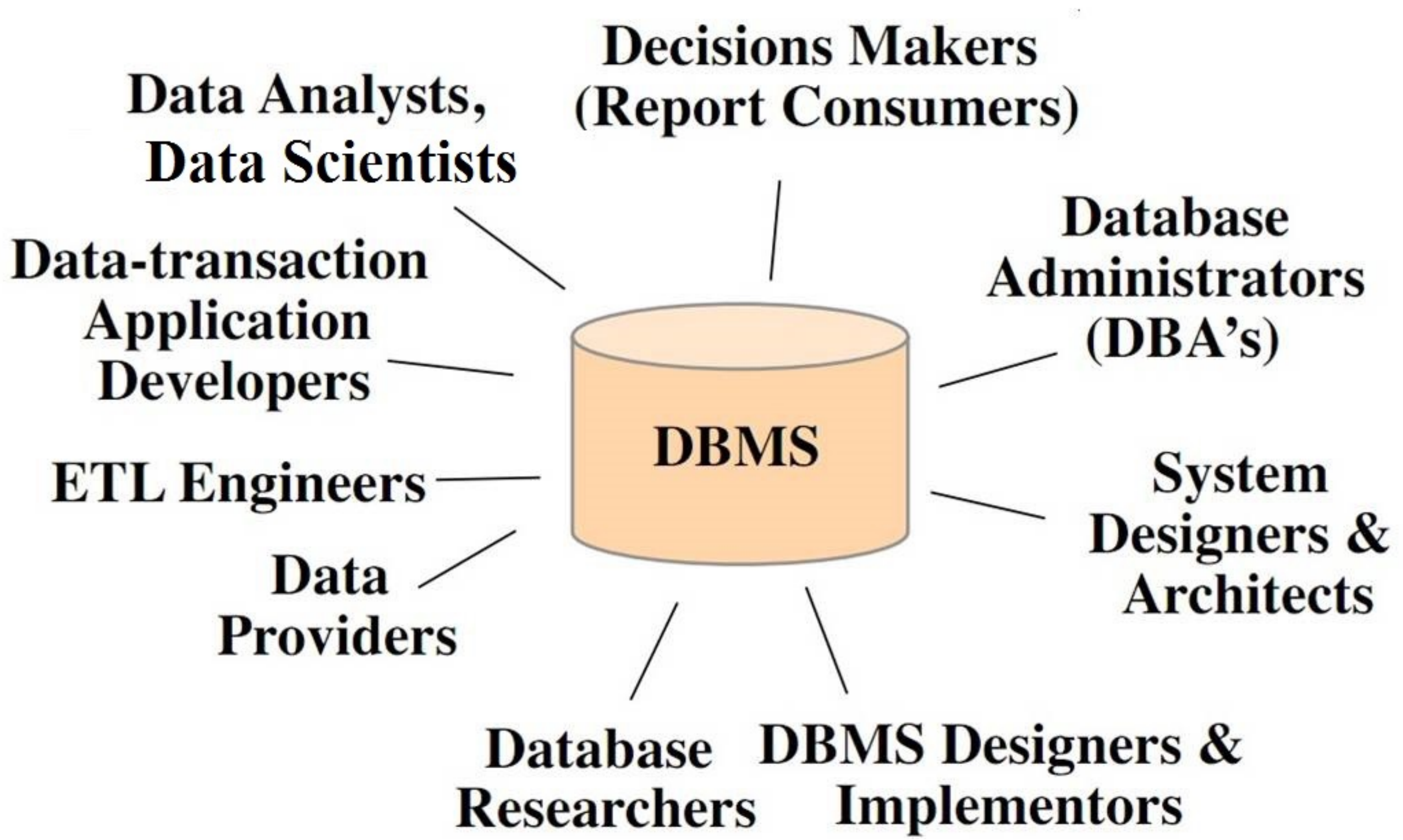
- 1980s
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - IBM, Teradata
  - Object-oriented database systems
- 1990s
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History of Database Systems (continued)

- 2000s
  - Big data storage systems
    - Google BigTable, Yahoo PNUTS, Amazon Cloud
    - "**NoSQL**" systems: key-values, graph DBMSs, etc.
  - Big data analysis: beyond SQL
    - Map-reduce and friends
- 2010s and beyond
  - SQL "**reloaded**"
    - SQL is a front end to Map-Reduce systems (e.g. Hive)
    - Massively parallel database systems
    - Logical Data Warehouses
    - SQL-based multi-core in-memory databases
    - Spark SQL
- 2020s
  - Multi-Cloud Data Warehouses
    - E.g. Snowflake
  - Distributed Transactional Databases ("NewSQL" databases)
    - E.g. YugabyteDB


Datavault
@Datavault_UK
www.data-vault.com
Data Vault Technology Landscape
Winter 2018/2019
Dashboard
Reporting Engine
Data Science Tools
Good Data
Transformation
Complex Event Processors
Scheduler
Cleaning
Data
Traditional SQL
Big Data Cluster
DevOps
Languages
Version Control
Platform
Platform Utilities
Qlik, sas, Apache Zeppelin, TIBCO Spotfire, pentaho, Google Data Studio, ORACLE Hyperion, Microsoft SQL Server Reporting Services, DOMO, tableau, Pivotal Greenplum, Yellowfin, COGNOS, birst, TIBCO Jaspersoft, MicroStrategy, Power BI, Information Builders, CHARTIO, SAP Business Objects, ORACLE Business Intelligence, QuickSight, informatica
Dashboards and Reporting Engines
spyder, KNIME, SciPy, C Language, SPSS, mahout, neopulse, python, MATLAB, sas, jupyter, R, TensorFlow, H2O.ai, databricks, GNU Octave
Data Science (including AI and ML)
Qubole, kafka, Amazon Kinesis, nifi, samza, alooma, StreamSets, RabbitMQ, TIBCO Spotfire, elasticsearch, FLUME, informatica, Apache STORM, Apache Flink, WS
Streaming
Microsoft SQL Server Integration Services, tamr, snaplogic, alooma, Fivetran, Dbvisit, ATTUNITY, MATILLION, alteryx, talend, StreamSets, DiYOTTA, IBM Datastage, Ab Initio, WS WhereScape, pentaho, CloverDX, Biml, DataBlade, informatica, BimlFlex
ETL/ELT
Waterline Data, Royal Mail, WS, alteryx, IBM Infosphere, tamr, IRI, Datamaran
Cleaning
talend, WS, alteryx, BimlFlex, cron, OOZIE
Scheduler
Reporting Systems
Integration
Redshift, amazon DynamoDB, cassandra, Google BigQuery, CouchDB, Amazon Aurora, MySQL, snowflake, mongoDB, teradata, MariaDB, ORACLE DATABASE, redis, hadoop HDFS, Amazon Athena, Microsoft SQL Server, PostgreSQL, IBM DB2, SAP S/4 HANA, INFOBRIGHT
Databases/File systems
XHTML, <xml />, {JSON}, CSV, Apache orc, COBOL, AVRO
Data Formats
HIVE, presto, sqoop, cloudera IMPALA, Apache Spark
SQL Engines
Database Systems
amazon web services, Windows Azure, Google Cloud Platform, rackspace
Platform
ANSIBLE, Jenkins, Planview LeanKit, CloudFormation, Jira Software, HashiCorp Terraform, COLLABNET VERSIONONE
DevOps
Apache KNOX, dynatrace, Apache Ranger, Apache FALCON, Ganglia
Platform Utilities
CVS, git, mercurial, subversion
Version Control
Scala, T-SQL, Java, python, ORACLE PL SQL
Backend Languages
Platform and DevOps
©Datavault is a trading name of Business Thinking Ltd. All Trademarks acknowledged. While we strive to make this information as accurate as possible, we make no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the contents, and expressly disclaims liability for errors and omissions.

Database Users: Who Needs to Use DBMS?


Decisions Makers (Report Consumers)
Data Analysts, Data Scientists
Data-transaction Application Developers
ETL Engineers
Data Providers
DBMS
Database Administrators (DBA's)
System Designers & Architects
Database Researchers
DBMS Designers & Implementors

Everybody who works with data!

Elements of DBMS Ecosystem

- **Data Models**
- Database Design
- Data Access
  - **DDL:** Data Definition Language
  - **DML:** for Data Manipulation Language
    - Both use SQL
- DBMS Engine
  - Query Processor
  - Transaction manager
  - Storage manager
- Database Architecture
- Database Users

Data Models

A collection of tools for describing

- Data
- Data relationships
- Data semantics
- Data constraints

Types of Data Models

- Relational model
- Entity-Relationship (E-R) data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (e.g. XML)
- Other older models:
  - Network model
  - Hierarchical model

Relational DBMS (RDBMS)

- Data stored in "tables" with relations between them.

Students

SID	Name
1	Alice
2	Bob
3	Charlie
4	Debra

Courses

CID	Name
1	ANLY640
2	ANLY502
3	ANLY503
4	ANLY511

Enrollments

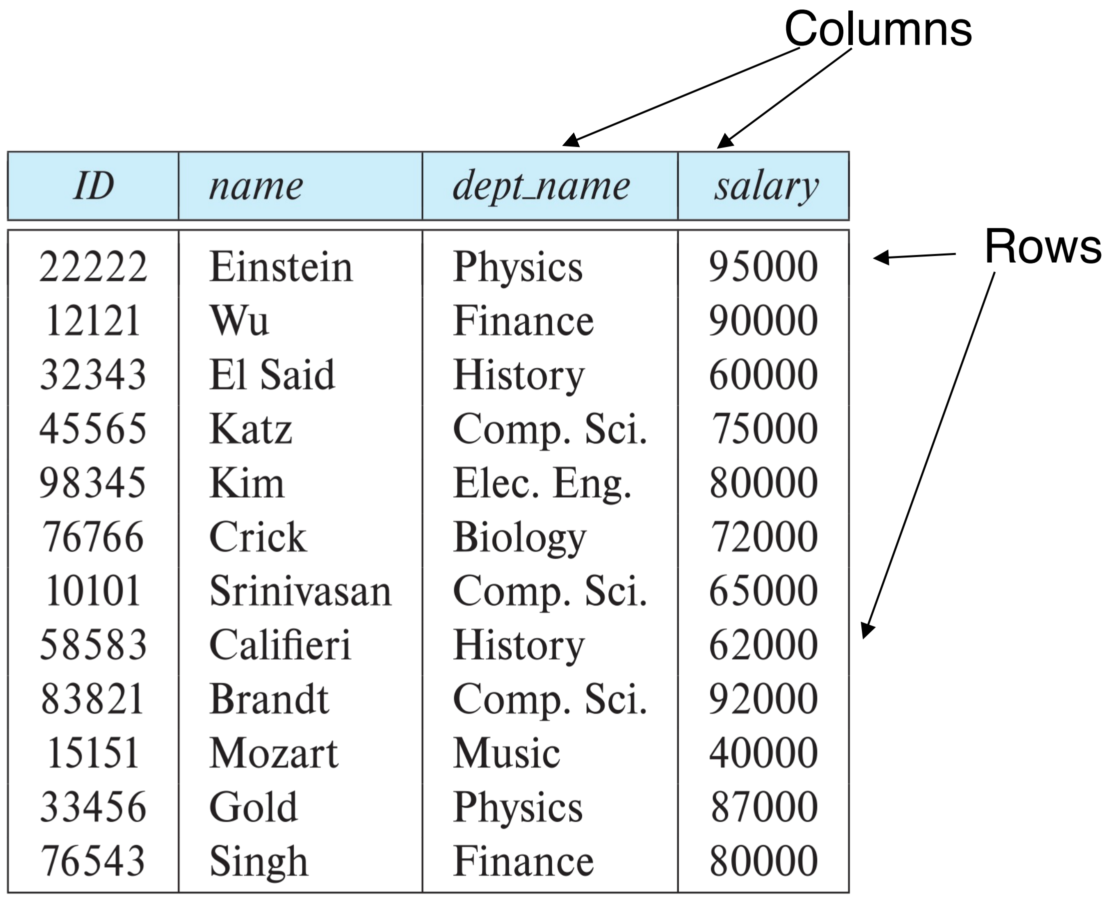
Key Foreign Keys

EID	SID	CID
1	1	1
2	1	2
3	2	1
4	3	2

- Alice is enrolled in ANLY640 and ANLY502
- Bob is enrolled in ANLY640
- Charlie is enrolled in ANLY502
- Debra is not enrolled in any course.

Relational Model

- All the data is stored in various tables, called relations.
- Example of tabular data in the relational model


Columns
Rows

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

Elements of DBMS Ecosystem

- Data Models
- **Database Design**
- Data Access (DDL and DML)
  - SQL
- DBMS Engine
  - Query Processor
  - Transaction manager
  - Storage manager
- Database Architecture
- Database Users

Database Design

The process of designing the general structure of the database usually involves the following steps

- Step 1: Requirements Analysis
- Step 2: Conceptual design
  - Database design requires that we find a “good” collection of relation schemas.
  - Goal: describe data and their relationships
- Step 3: Logical Design
  - Involves designing of the database schema.
  - Start of Implementation phase
- Step 4: Physical Design
  - Deciding on the physical layout of the database

Requirements Challenge

- Data is collected much faster than it can be transformed into valuable information.
- Needed: identify the nuggets of information that is worth extracting from the avalanche of data – and what can be ignored.

- Catch 22:
  - IT department needs requirements to build a database
  - Customers often need to first understand the data in order to provide these requirements, which might not be possible until IT department builds an environment for them to use

This is very important… Unimportant, of course, I meant—important, unimportant, unimportant, important.
Lewis Carroll's Alice's Adventures in Wonderland

Instances and Schemas

- Similar to types and variables in programming languages
- **Logical Schema** – the overall logical structure of the database
  - Example: University database consists of information about a set of students and courses in a university and the relationship between them
- **Physical schema**– the overall physical structure of the database, e.g. on disk
- **Instance** – the actual content of the database at a particular point in time (including data)

Elements of DBMS Ecosystem

- Data Models
- Database Design
- **Data Access (DDL and DML)**
  - SQL
- DBMS Engine
  - Query Processor
  - Transaction manager
  - Storage manager
- Database Architecture
- Database Users

Data-Definition Language (DDL)

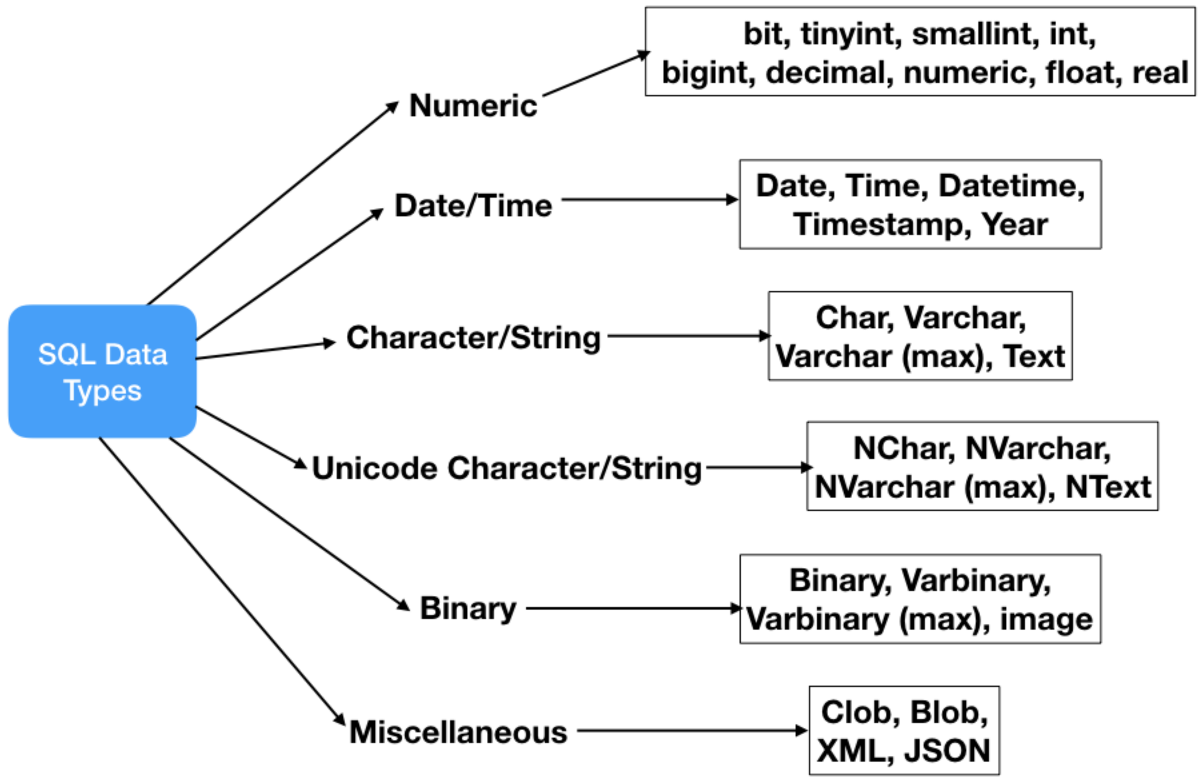
- Specification notation for defining the database schema

  Example:

  ```
  create table instructor (
      ID          char(5),
      name        varchar(20),
      dept_name   varchar(20),
      salary      numeric(8,2))
  ```

- DDL compiler generates a set of templates stored in a **Data Dictionary**
- Data dictionary usually contains:
  - Metadata (i.e., data about data)
  - Database schema
  - Constraints, e.g.
    - Primary key (e.g. ID uniquely identifies instructors) for referential integrity
    - User-defined (e.g. budget > 0)
  - Authorization: Who can access what

Example of SQL Datatypes


SQL Data Types
Numeric
bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
Date/Time
Date, Time, Datetime, Timestamp, Year
Character/String
Char, Varchar, Varchar (max), Text
Unicode Character/String
NChar, NVarchar, NVarchar (max), NText
Binary
Binary, Varbinary, Varbinary (max), image
Miscellaneous
Clob, Blob, XML, JSON

https://www.journaldev.com/16774/sql-data-types

Data Manipulation Language (DML)

- Language for accessing and updating the data
  - Allows to retrieve, insert, delete and update data organized by the appropriate data model
- There are two types of data-manipulation language
  - **Procedural DML** -- require a user to specify what data are needed and how to get those data.
  - **Declarative DML** -- require a user to specify what data is needed without specifying how to get that data.
- The portion of a DML that involves information retrieval is called a **query** language.
  - Though sometimes query language term is applied to all DML functions

Elements of DBMS Ecosystem

- Data Models
- Database Design
- Data Access (DDL and DML)
  - **SQL**
- DBMS Engine
  - Query Processor
  - Transaction manager
  - Storage manager
- Database Architecture
- Database Users

SQL

- SQL is a standard language for querying and manipulating data

- SQL is a Declarative language
  - *What* but not how

SQL stands for
Structured **Q**uery **L**anguage

- Many standards out there:
  - ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3), ….SQL:2016 (aka SQL8)
  - Vendors support various dialects and extensions

Probably the world's most successful parallel programming language

Adapted from : http://web.stanford.edu/class/cs145/

SQL (continued)

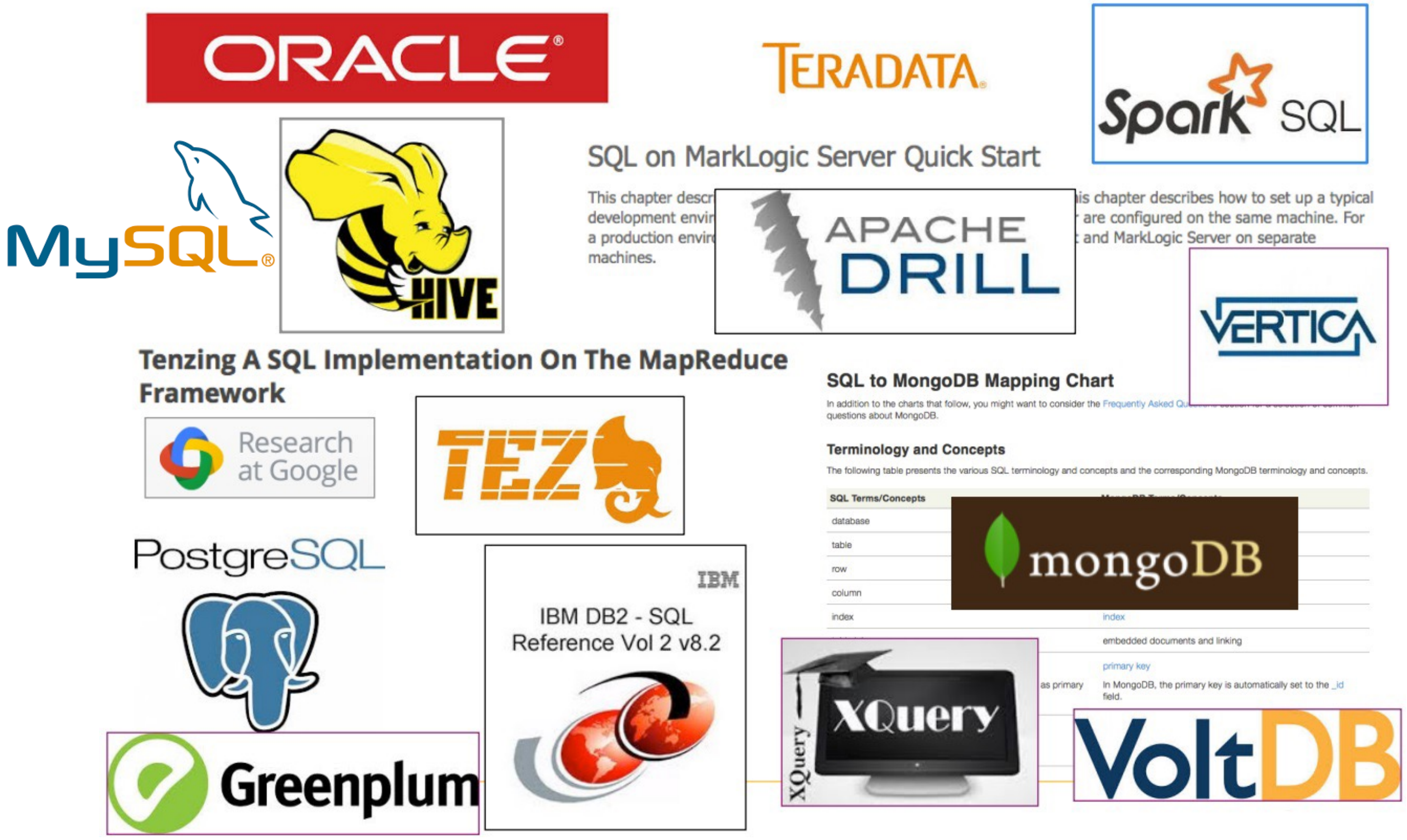
- SQL commands — designed to read like English.

- Examples of SQL Commands are:
  - CREATE — Creates a database or table
  - SELECT — Reads data
  - INSERT — Inserts new data
  - UPDATE — Changes existing data

- Example: to find all instructors in Comp. Sci. dept

  select *name*
  from *instructor*
  where *dept_name* = 'Comp. Sci.'

SQL Touches Every Piece of Modern Technology



SQL on MarkLogic Server Quick Start

This chapter descr… development envir… a production envir… machines. …is chapter describes how to set up a typical … are configured on the same machine. For … and MarkLogic Server on separate

Tenzing A SQL Implementation On The MapReduce Framework

IBM DB2 - SQL Reference Vol 2 v8.2

SQL to MongoDB Mapping Chart

In addition to the charts that follow, you might want to consider the Frequently Asked Qu… questions about MongoDB.

Terminology and Concepts

The following table presents the various SQL terminology and concepts and the corresponding MongoDB terminology and concepts.

SQL Terms/Concepts	
database	
table	
row	
column	
index	index
	embedded documents and linking
	primary key
as primary	In MongoDB, the primary key is automatically set to the _id field.

Application Programs

- Non-procedural query languages such as SQL are not as powerful as a universal Turing machine.
- SQL does not support actions such as input from users, output to displays, or communication over the network.
- Such computations and actions must be written in a host language, such as C/C++, Java or Python, with embedded SQL queries that access the data in the database.
    - **Application programs** -- are programs that are used to interact with the database in this fashion.

Elements of DBMS Ecosystem

- Data Models
- Database Design
- Data Access (DDL and DML)
  - SQL
- **DBMS Engine**
  - Query Processor
  - Transaction manager
  - Storage manager
- Database Architecture
- Database Users

DBMS Engine

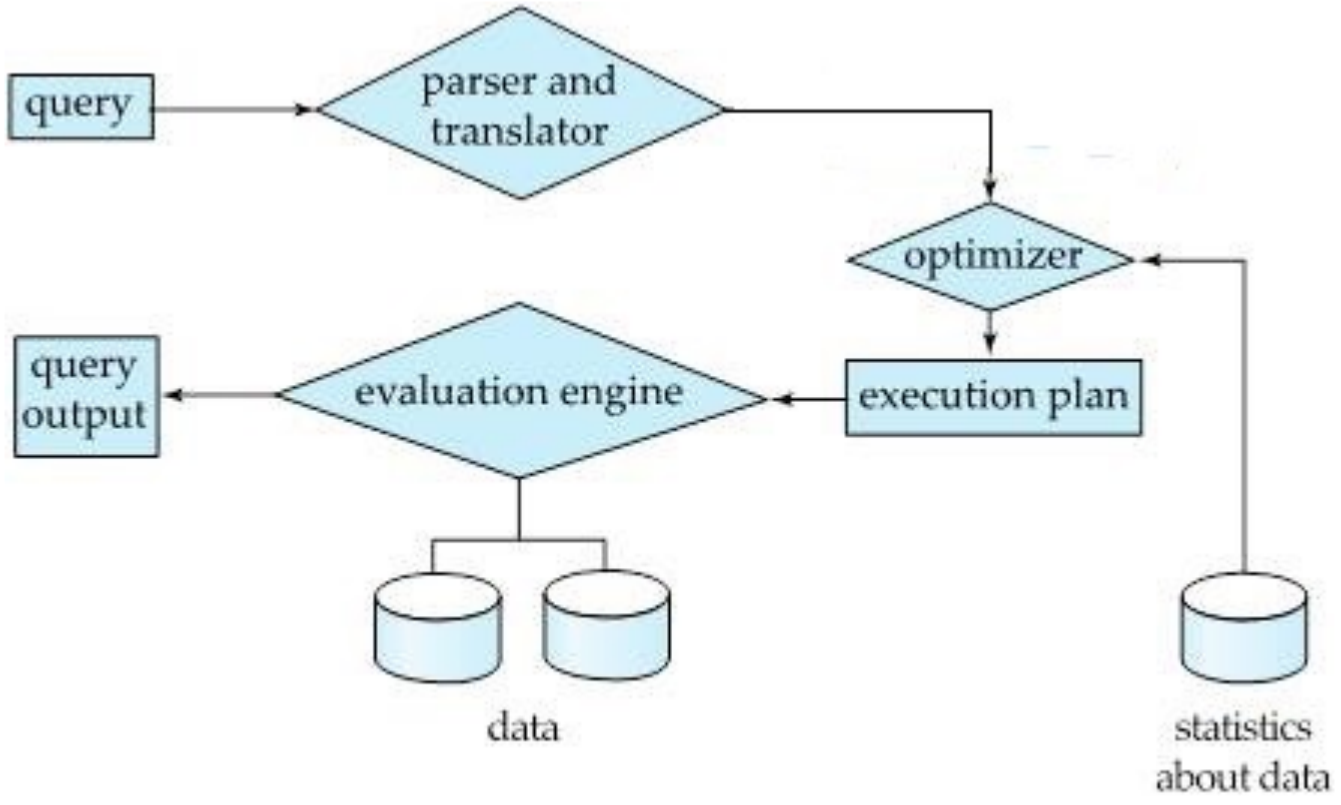
- A database system is partitioned into modules that deal with each of the responsibilities of the overall system.
- The functional components of a database system can be divided into:
  - **Query processor**
    - Helps to accelerate database processing
    - Can be done in parallel if more than one node/computer
  - **Transaction manager**
    - Allows users and developers to treat a sequence of db accesses as a single one
    - Manages concurrent access
  - **Storage manager**
    - Especially important for Massive Data storage
    - Can be done in parallel if more than one node/computer

DBMS Query Processor

- The query processor components include:
  - <u>DDL interpreter</u>
    - Interprets DDL statements and records the definitions in the data dictionary.
  - <u>DML compiler</u>
    - Translates DML statements in a query language into an execution plan consisting of low-level instructions that the query evaluation engine understands.
    - The DML compiler performs query optimization; that is, it picks the lowest cost execution plan from among the various alternatives.
  - <u>Query evaluation engine</u>
    - Executes low-level instructions generated by the DML compiler.

DML Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



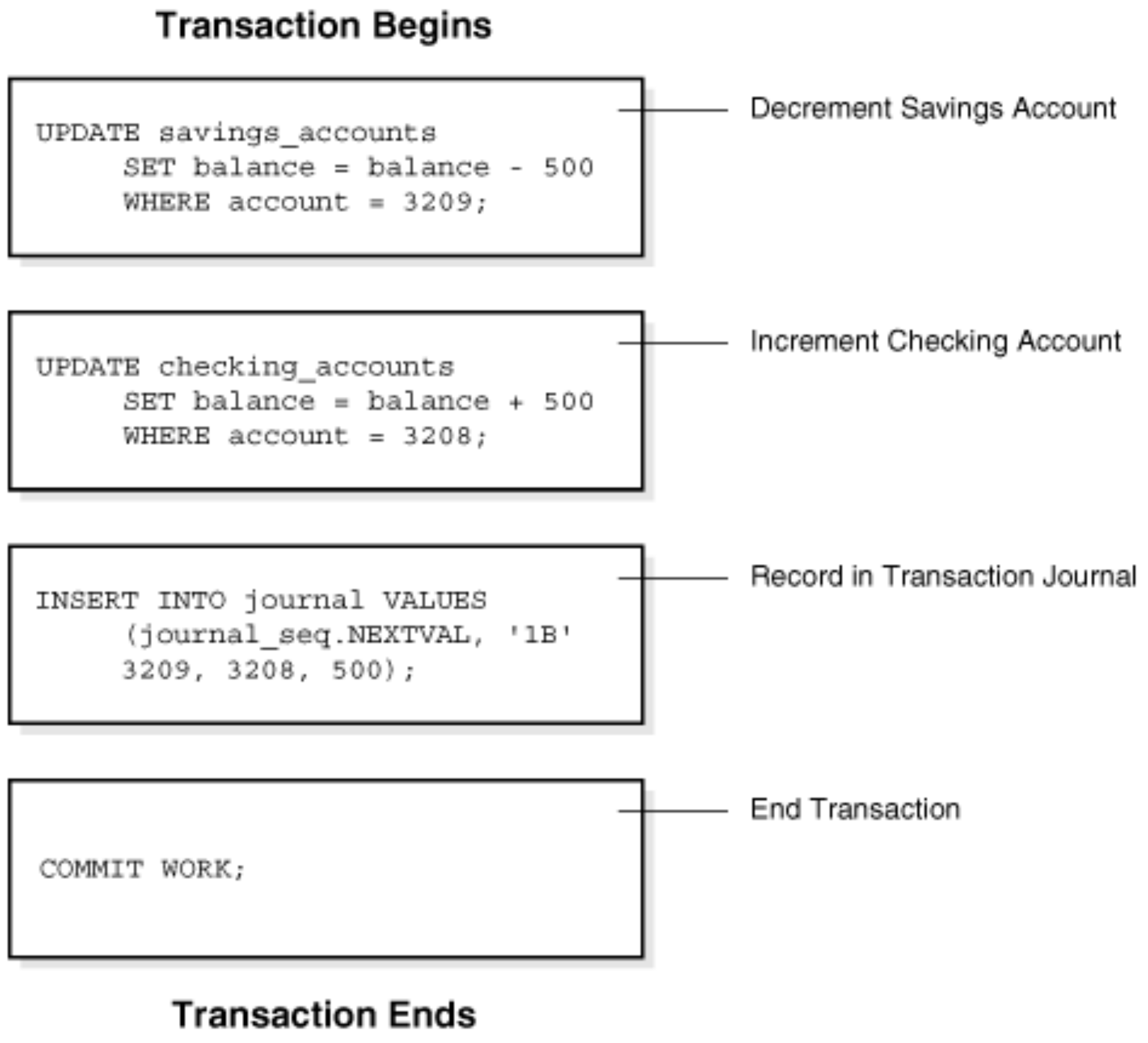
Elements of DBMS Ecosystem

- Data Models
- Database Design
- Data Access (DDL and DML)
  - SQL
- DBMS Engine
  - Query Processor
  - **Transaction manager**
  - Storage manager
- Database Architecture
- Database Users

Transaction Manager

- **Transaction** is a collection of operations that performs a single logical function in a database application
- **Transaction-management component** ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- **Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Example of a Transaction


Transaction Begins
Decrement Savings Account
Increment Checking Account
Record in Transaction Journal
End Transaction
Transaction Ends

```
UPDATE savings_accounts
       SET balance = balance - 500
       WHERE account = 3209;
```

```
UPDATE checking_accounts
       SET balance = balance + 500
       WHERE account = 3208;
```

```
INSERT INTO journal VALUES
       (journal_seq.NEXTVAL, '1B'
       3209, 3208, 500);
```

```
COMMIT WORK;
```

Elements of DBMS Ecosystem

- Data Models
- Database Design
- Data Access (DDL and DML)
  - SQL
- DBMS Engine
  - Query Processor
  - Transaction manager
  - **Storage manager**
- Database Architecture
- Database Users

Storage Manager

- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - Interaction with the OS file manager
  - Efficient storing, retrieving and updating of data
- The storage manager components include:
  - Authorization and integrity manager
  - File manager
    - Manages allocation of space on disk
  - Buffer manager
    - Fetches data from disk into memory

Storage Manager (continued)

- Raw data is stored on disk using the file system provided by OS
- The storage manager implements several data structures as part of the physical system implementation, for example:
  - Data files -- the database itself
  - Data dictionary -- metadata about the structure of the database, in particular, the database schema.
  - Indices -- can provide fast access to data items.
    - A database index provides pointers to those data items that hold a particular value.

Elements of DBMS Ecosystem

- Data Models
- Database Design
- Data Access (DDL and DML)
  - SQL
- DBMS Engine
  - Query Processor
  - Transaction manager
  - Storage manager
- **Database Architecture** (will start here next class)