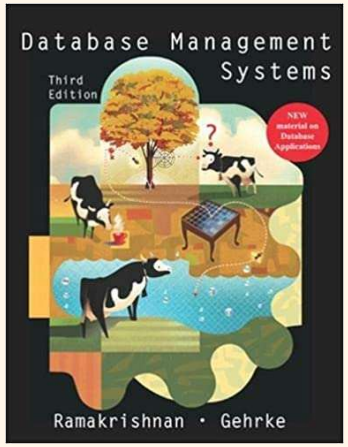
CPSC 404:
Advanced Relational Databases

Section 201 (9:30-10:50 T/R)

Instructor: Dr. Ed Knorr (he/him)
Department of Computer Science
University of British Columbia



January-April 2026

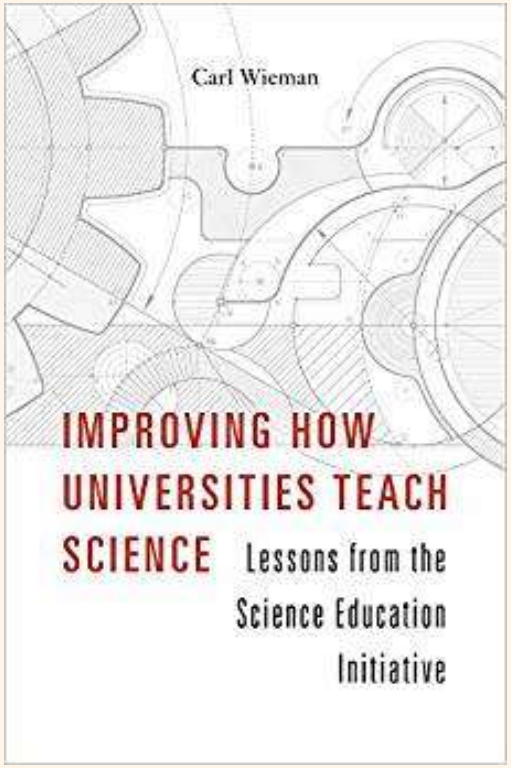
- Please read the Course Outline (syllabus) carefully.
- I'll keep it updated on Canvas.
- I'll make an announcement in class, or I'll post a note, when any significant change has been made.

Learning Goals for This Unit

- List important job responsibilities of various DB personnel.
- Explain the benefits of logical and physical data independence brought about by the relational model.
- Explain why DBMSs are so hard to configure properly for an organization.
- Identify some database challenges in providing 24×7 operations.
- Justify the use of self-managing (sometimes called "autonomic" or self-tuning) database systems.
- Provide examples of DBA (database administration) tasks that can be simplified using autonomic computing.
- Give examples of quantifiable components that determine the "all-in cost" of delivering database services.
- Give examples of queries that take good advantage of hash-based indexes, and of tree-based indexes.
- With examples of queries, explain the difference between clustered and unclustered tree-based indexes.

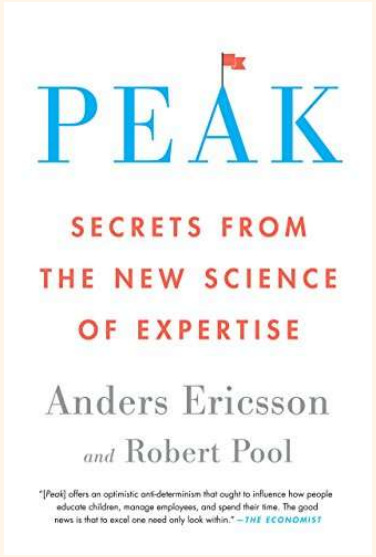
Skylight and CWSEI

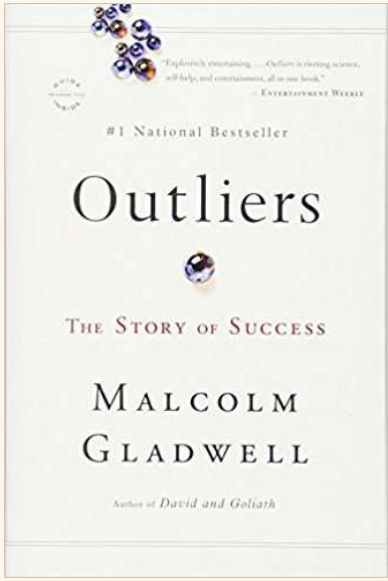
- Part of the Faculty of Science's mandate to improve teaching and learning in Science
  - Best practices from evidence-based research
  - Skylight = Science Centre for Learning and Teaching (SCLT but pronounced "Skylight")
  - CWSEI = Carl Wieman Science Education Initiative (pronounced C-W-see)
  - www.cwsei.ubc.ca has some great resources; you may wish to explore the site, and especially the *Resources* page and its links



Skylight and CWSEI (cont.)

- In this course, there will be an emphasis on deliberate practice and problem solving.
  - What is "deliberate practice"?
  - We're want students to develop expert-like thinking about a task.
  - We'll use clickers, analogies, worked examples, in-class exercises, more frequent testing, etc.
- Just about any world-class "expert" at a complex profession has spent about 10,000 hours of deliberate practice to achieve this level.
  - Beatles, Mozart, Tiger Woods, Connor McDavid, Bill Gates, scientists, physicians/specialists, etc.





Skylight and CWSEI (cont.)

- Learning Goals — very important!
  - Less of a focus on memorizing facts, and more of an emphasis on: applying concepts and problem solving
  - Bloom's Taxonomy (with a % estimate of typical exam questions in first-year, post-secondary, science courses, shown below)
    - However, CS tends to have more questions in levels 3-6 than other sciences.

Level #	% Questions	Level/Category	Some Verbs Used for Questions
6 (highest)	< 2%	Evaluation	Evaluate, Justify, Argue, Judge, …
5	< 2%	Synthesis	Create, Generate, Propose, …
4	2-5%	Analysis	Analyze, Determine, …
3	12-15%	Application	Apply, Use, Choose, …
2	20%	Comprehension	Explain, Show, Extrapolate, …
1 (lowest)	60%	Knowledge	Describe, List, Name, Identify, …

Bloom's Taxonomy

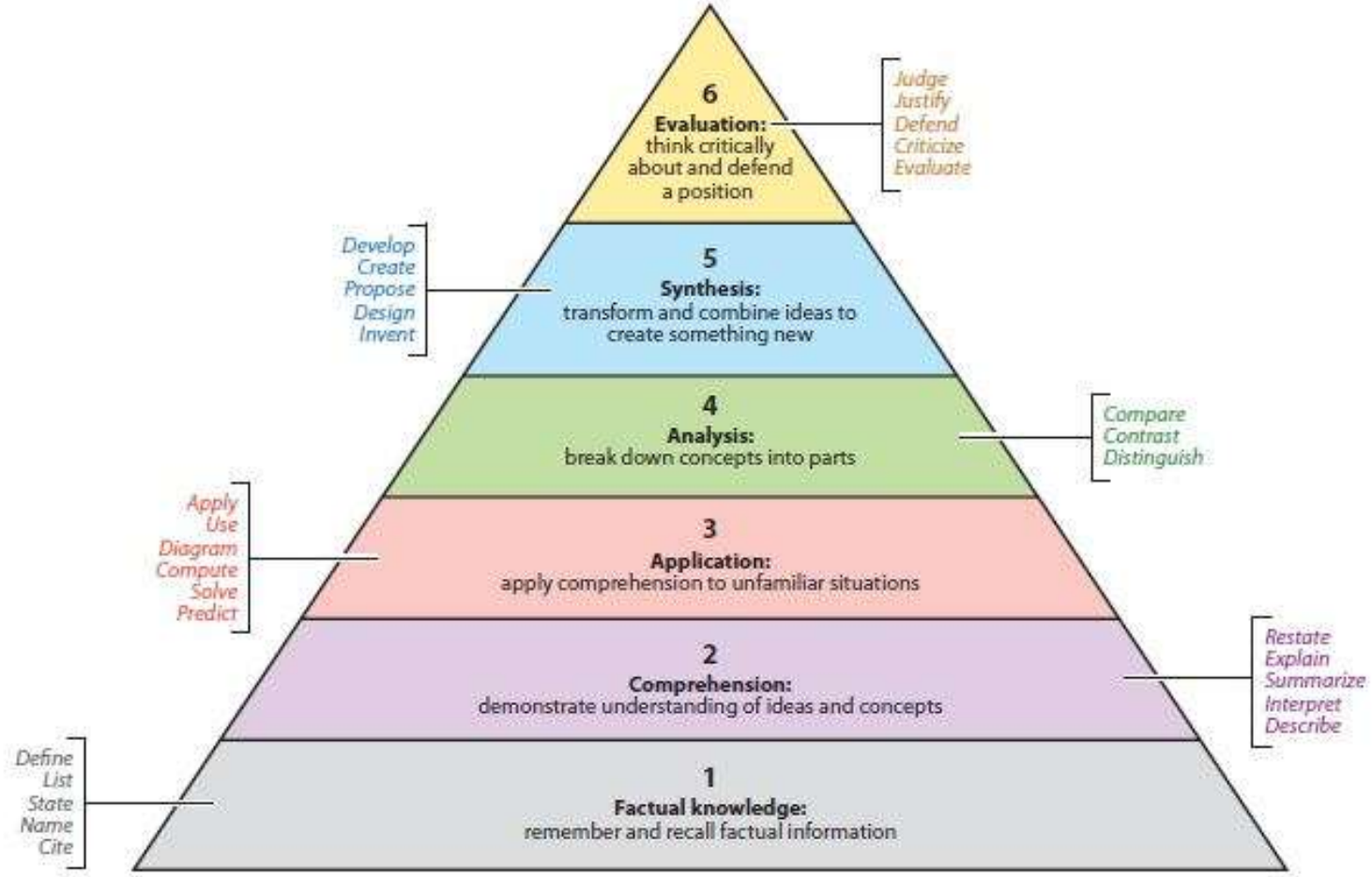


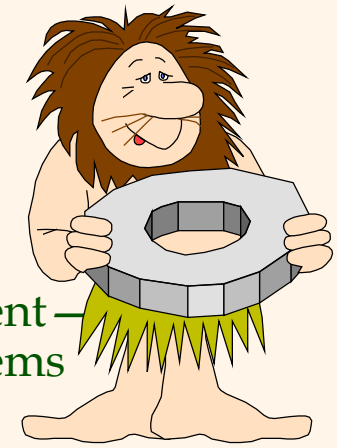
Figure 1

Bloom's levels of understanding. Originally termed Bloom's taxonomy of the cognitive domain, this schema defines six levels of conceptual understanding according to the intellectual operations that students at each level are capable of (Bloom & Krathwohl 1956). The italicized verbs have been added to the original hierarchy; they indicate performance tasks that test achievement of learning goals at each level. Fine distinctions in the hierarchy are difficult, and some educators prefer to classify goals on only three levels: low (1, 2), medium (3, 4), and high (5, 6). (Based on Allen & Tanner 2002.)

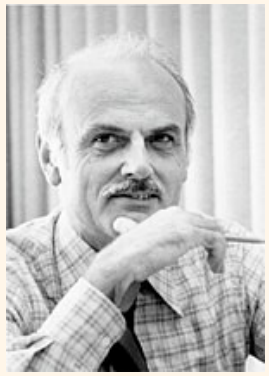
Course Objectives: We'll Study …

- The role of an RDBMS in an organization's data management strategy.
- The relationship among bytes, pages, disks, buffer pools, data tables, indexes, metadata, etc.
- Indexing strategies
- SQL query evaluation and optimization
  - Performance issues
- Transaction Processing and Concurrency Control
  - Schedules, Serializability, Deadlock, Locking Protocols, Isolation Levels
- Crash Recovery and Application Recovery
  - Logging
  - Backups (Image Copies) and Recoveries

Historically, 3 Major Types of DBMSs



- Hierarchical DBMSs (late 1960s)
  - e.g., IBM's IMS (Information Management System)
  - Many commercial applications have natural hierarchies in their data.
  - Many production systems throughout the world (e.g., banking, government – Government of Ontario), but nowhere near as many as for relational systems
- Network DBMSs (1970s)
  - e.g., IDMS from Computer Associates (CA)
  - Has lots of pointers (directed graphs), similar in spirit to the WWW
  - More difficult to program, less flexible
- Relational DBMSs (RDBMSs, late 1970s)
  - IBM's DB2 (rebranded to "Db2" in 2017), Oracle, Microsoft's SQL Server, MySQL, etc.
  - Heavily used throughout the world!
  - Bruce Lindsay, IBM: "Relational databases form the bedrock of Western civilization".
  - The focus of this course



RDBMSs and Beyond

- RDBMSs (cont.)
  - They are a workhorse, containing a lot of the data in the Deep Web (of which 96% is estimated to be generally inaccessible to search engines).
  - They spend a lot of time keeping data *consistent*, and making sure the data obeys the schema rules.
    - e.g., domain constraints, foreign keys, and uniqueness are a few examples
  - They may have limitations on size and performance.
    - e.g., answering mobile queries from a large number of users
- Object-Oriented DBMSs
  - They haven't really caught on, at least not the way some people had predicted
  - Very small market share (e.g., 1%)

RDBMSs and Beyond (cont.)

- For a good, readable article on the history and development of DBMSs, see the paper "What Goes Around Comes Around" by Mike Stonebraker and Joe Hellerstein.
- They cover 9 data model proposals or eras:
  - Hierarchical
  - Network (Directed Graph)
  - Relational
  - Entity-Relationship
  - Extended Relational
  - Semantic
  - Object-Oriented
  - Object-Relational
  - Semi-Structured (e.g., XML)

Newer Data Management Technologies: NoSQL

- NoSQL
  - "Not Only SQL" (or in some circles: No SQL at all)
    - They're non-relational systems.
    - They support distributed data, easily. Consider the idea of hash functions or load balancing that determine which machine the data gets sent to/from.
      - Sharding
      - Replication
  - Under the NoSQL banner, we have these and more:
    - Key-value stores
    - Column stores
    - Document stores
    - Graph databases
    - XML object-based stores
  - Examples of NoSQL systems include Cassandra, Apache CouchDB, DynamoDB (from Amazon), HBase, MongoDB, neo4j, and Redis

Newer Technologies: NoSQL (cont.)

- NoSQL often has an "eventually consistent" capability rather than the much stronger ACID properties of Atomicity, Consistency, Isolation, and Durability
- There's the lack of a schema.
  - Advantage?
  - Disadvantage?
- Big Data often deals with semi-structured data, or data without structure, including sensor data, images, audio, etc.
- Major objective of NoSQL systems: availability
- NoSQL often has denormalized data, and the systems typically don't do much in the area of *join* operations. RDBMSs do lots of joins.

Newer Technologies: NoSQL (cont.)

- CAP Theorem: Pick a maximum of 2 of: Consistency, Availability, and Partition-Resistant
  - Analogous to many real-world situations including some retail products, dining, services, construction, etc:
    - "Cheap, fast, and good: pick at most two"
- Documents are often represented using the JSON standard (JavaScript Object Notation).

Some DB-Related Jobs

- DB-related staff in a large IT shop (e.g., a major corporation) or as part of a consulting company:
  - Systems / Technical Support
    - Installation, configuration, customization, problem determination, interaction with vendor, patches, monitoring, performance & tuning, availability, disk storage, capacity planning, chargeback, etc.
  - Database Administration (e.g., DBA = Database Administrator or Database Analyst)
    - Requirements analysis, data modeling (logical DB design including Entity-Relationship Diagrams), data dictionary, standards, documentation, physical DB design, capacity planning, etc.
    - SQL analysis (e.g., identify performance bottlenecks, recommend changes, add indexes)
    - Support business cases; interact with users, programmers, technical support, management, business partners, and other stakeholders

Some DB-Related Jobs (cont.)

- DB-related staff (cont.)
  - Database Support (often combined with Database Administration as part of a DBA's role)
    - Set up production jobs including backups and recoveries, utilities, authorizations, binds/rebinds, etc.
    - Set up and load DBs in test and production environments
    - Interact with other DB and technical staff
    - Nightly and weekend callouts are not uncommon
  - Programming
    - Historically, each area within a corporation had its own programming group (e.g., for an oil & gas company: payroll, human resources, accounting, land, drilling, exploration, production, …)
    - DBA interacts with each group regularly

Some DB-Related Jobs (cont.)

- Software Vendors
  - They write and maintain a DBMS.
    - e.g., Oracle, IBM, Microsoft, …

  - Staff include:
    - Scientists, designers, programmers, analysts, testers, etc.
    - Sales and marketing, product support, training, etc.
    - Later in the course: Each vendor may use different ways of implementing of B+ trees and other indexes, different ways of handling overflows and clustering, different page/record layouts, different optimizer algorithms, different recovery systems, etc.
      - In this course, we'll have lots of pedagogical (learning) DB principles, and look at some of the varieties of implementation.
      - Your employer—if you get a programming job with a DB vendor—will provide all the details for their system.

Some DB-Related Jobs (cont.)

- Application/Middleware Vendors
  - They write software that integrates with a DBMS, and which many companies purchase to help support their DBMS.
  - Some vendors specialize in HR software, financial services, data warehouses for a given industry (e.g., airlines), etc.
- Thus, there are many DB-related jobs.
  - Some large IT consulting shops employ people in each of the categories on the previous 2 pages to service multiple clients, although some customers are so big that multiple consultants are often devoted to—and permanently stationed at—a single client's site.
  - In a small shop, a single employee is often responsible for many of the above tasks.
    - Do you want to be a "jack of all trades"—or a specialist?
    - A good way to decide is via co-op jobs with different employers.

Outsourcing

- Companies often save money by outsourcing their IT departments (or major parts of it).
  - e.g., BC Hydro → Westech Information Systems and Western Integrated Technologies → BC Hydro → Accenture and others

  - e.g., Does each corporation really need to develop, maintain, and support *its own* proprietary payroll, HR, and tax programs?
    - e.g., BC Hydro moved to → PeopleSoft HR software → later Oracle buys PeopleSoft
    - e.g., airlines, airports, baggage handling, etc.
    - e.g., For decades, UBC used its own home-grown systems for HR, payroll, student records, registration, etc. → now, Workday.

Outsourcing (cont.)

- Software, hardware, and networks change frequently; networks, mobile solutions, and security are "moving targets".
- Some multinationals are consolidating operations and working from a distance (e.g., Shell Bangalore).

- Consider software as a service (SaaS), database as a service (DBaaS), and cloud storage (e.g., Amazon Web Services, Microsoft Azure, Google Cloud Platform).
  - Examples of such products (courtesy of Nerdio, 2016):
    - Salesforce.com
    - Microsoft Office 365
    - Box
    - Google Apps
    - Amazon Web Services
    - MongoDB and Atlas – can use AWS, GCP, or Azure
    - Zendesk
    - Docusign
    - Dropbox
    - Slack

Outsourcing (cont.)

- The all-in cost of an employee is approximately double his/her salary. Why?
  -
  -
  -
  -
  - Training costs
    - e.g., $3000-5000 for a 5-day DB tuning course (or DB conference) in Toronto, *plus* one-week's loss of an employee
- There are trade-offs to outsourcing:
  - costs, security, privacy, stability, knowledge of the business area, competitiveness, time to respond
  -
  -
  -

Data Independence

- Joe Hellerstein's Inequality:
  - *d* application / *dt* << *d* environment / *dt*
  - What are the implications for applications that use databases?
- From CPSC 304, recall some major benefits of a relational DBMS:
  - Sharing, Redundancy, Integrity, Concurrency, Backup, Recovery
  - Physical independence (indexes, reorgs, replication, disks, other hardware)
  - Logical independence (schemas, add/drop fields, no pointer manipulation, a declarative query language)
- CPSC 304 dealt with usage of a DBMS; in CPSC 404, we look under the hood (i.e., at the internals)

Availability & Performance

- Here are some desirable characteristics of a DBMS to permit high-levels of availability (e.g., 24×7) and performance:
  - Perform a schema change *without taking the database offline.*
    - Note that there could be complex changes that affect queries, updates, application programs, optimizations, backup/recovery, etc.
    - Creating a "schema evolution tool" is a very hard problem.
      - Important area of research
  - Online backup (i.e., while the data is still "live")
  - Online reorganization (of a table and its indexes)
  - Add/drop indexes on-the-fly.
  - Optimize the buffer pool to help minimize the number of disk I/Os.
  - Be proactive about poorly performing queries or resource hogs.

Self-Managing DBMSs

- Skilled DBAs are hard to find, especially in small shops where a single person cannot be a jack of all trades.
  - Performance and tuning are important DBMS tasks that require substantial know-how.
  - DBAs tend to work longer hours than most IT people, including some evenings and weekends.
- Many DBMSs have hundreds of tuning parameters, such as:
  - Checkpoint frequency, log size, sizes and types of buffer pools, rollback wait time, how often to check for deadlocks, max # of connections, max # of concurrent users, lock escalation levels, sizes of work files, # of concurrent users, security issues, etc.
  - Hard: What *defaults* should be provided to DBMS customers?
- So … DBMS vendors are putting more effort into self-managing database systems
  - e.g., IBM's "Autonomic Computing" initiative

Self-Managing DBMSs (cont.)

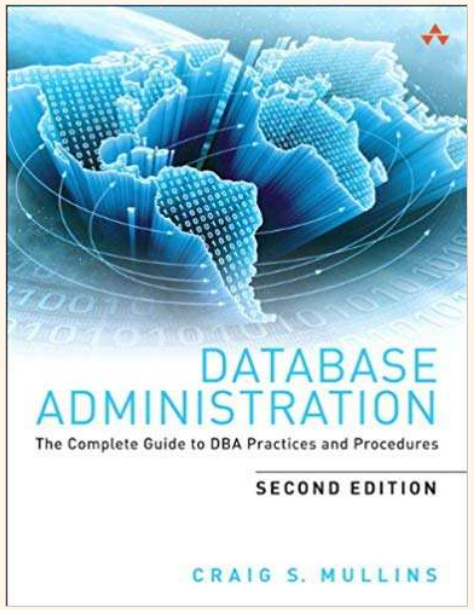
- Example 1: When should tables be backed up?
  - Common Practice:
  - Preferably:
- Example 2: When should tables be reorganized, and indexes be rebuilt?
  - Common Practice:
  - Preferably:
- A big part of this course is about performance.
  - How do we quantify "good performance"?
  - How do we design for good performance?
  - How can we fix bad performance?

Self-Managing DBMSs (cont.)

- Example 3: How do we reorganize a table and its indexes?
  - Traditionally, a reorg involved taking the database or table offline (making it unavailable to users), unloading the data, sorting it, reloading the data, rebuilding its indexes, capturing metadata, rebinding application plans, and bringing the data objects back online.
  - Better:
    - Make a duplicate of the data and reorganize the duplicate, while keeping the original data online for read and write access.
    - When the copy is reorganized, use the database log to *apply the latest changes* (that were made to the production data) *to the new copy*.
      - Iterate this update process, until there are no more changes.
    - Switch the two sets of objects.
- There are lots of vendor-supplied or third-party tools to help a DBA.

Availability

- Experienced DBA and author Craig Mullins wrote:
  - A major DBMS vendor reports that 70% of the trouble calls they receive about database outages are because of DBA mistakes.
  - Planned outages represent up to 70% of downtime.
  - Of the remaining 30% of downtime, up to half of those unplanned outages are due to problems encountered during the *planned* downtime.



Availability (cont.)

- To help keep systems running around the clock, use automated DBA tools as much as possible.
- Be sure to have well-trained and experienced staff.
- Use clusters of machines, network attached storage (NAS), and storage area networks (SANs).
- Standby systems:
  - Make use of redundant hardware with copies of DB log changes being propagated to other systems, ready to take over, if needed
  - Provide fast hardware failover support
  - Enable you to take a machine offline for service while the DBMS continues to run

Closing Thoughts

- Databases are critical parts of an organization's data assets.
- Much of the data on the Deep Web is stored in databases which are (usually) inaccessible to search engines.
- RDBMSs have stood the test of time—and have a great future.
- The Fortune 500 companies use RDBMSs.
- Many corporations have several RDBMSs (from different vendors).
- DB-related jobs are worth considering for your future.
- DB *principles* (e.g., indexing, transactions, concurrency, locking, contention, metadata, logging, backup, recovery, security, and the I/O cost model) are worth knowing, even if you don't plan to make databases part of your career.