**Database Design & Administration - CST8250**

**Week 5 - February 8th**

Indexes and Views

Class Intro

* We might be going back on campus for final exams
* The midterm is coming up on February 22nd and will be conducted during class time
* Indexes/views is the point where design meets the physical

Indexes

* Purpose of Indexes
* Indexes defined
* Index file organization
* Types of indexes
* Rules for indexes
* SQL examples for indexes

Views

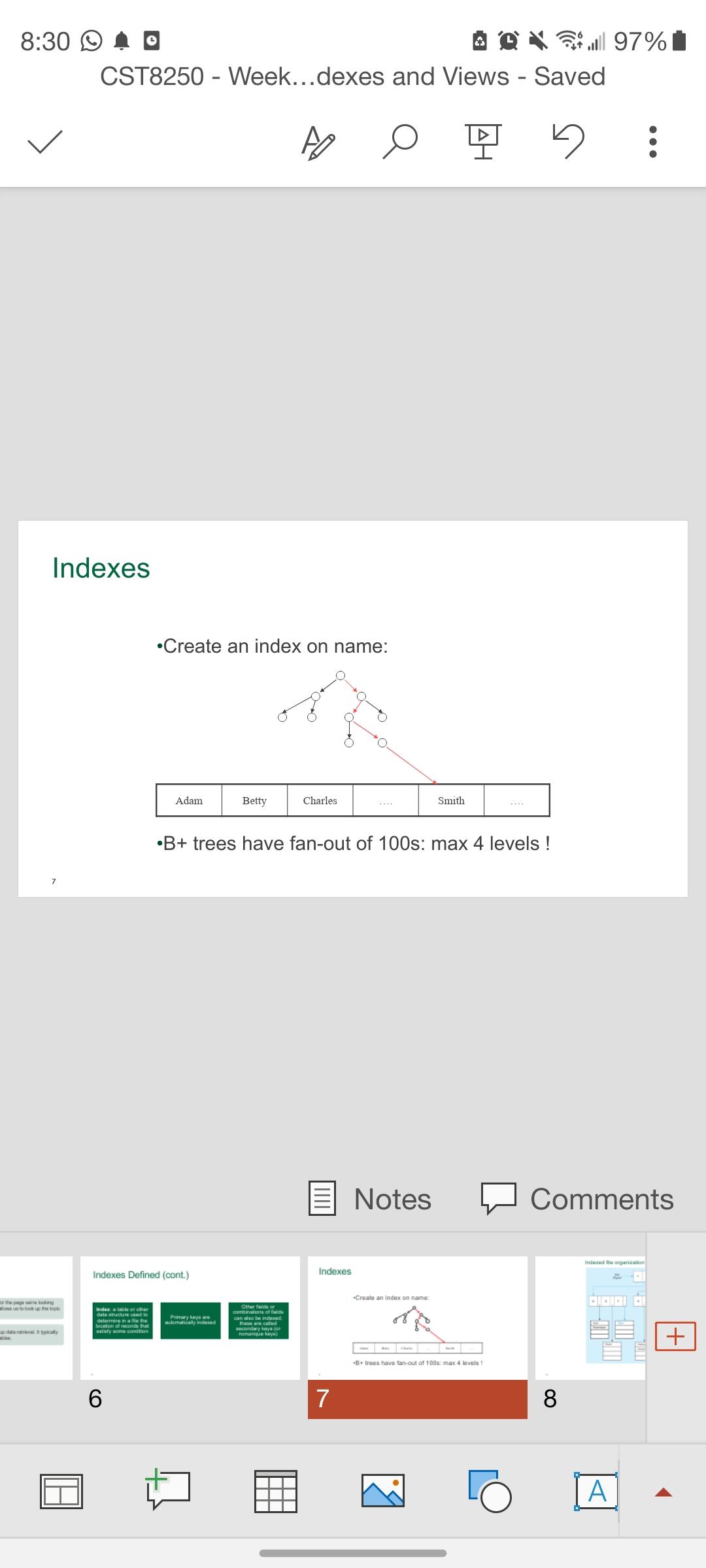
* Views defined
* View example
* Types of views
* SQL examples for views

Purpose of indexes

* Most queries only require a small amount of information from a database
  + Rarely do you do select \* from orders
* What if the only way to get the information was to search the ENTIRE database – INEFFICIENT!
* Indexes help us speed up our queries so we don’t have to search the entire database

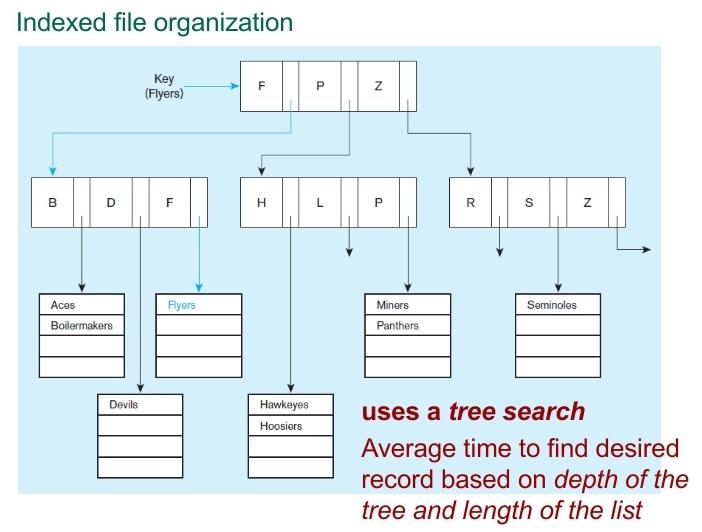
Indexes defined

* Analogy: Instead of searching an entire textbook for the page we’re looking for, we have an index at the back of the book that allows us to look up the topic and page number.
* Definition: A data structure that is used to speed up data retrieval. It typically contains a list of keys used to identify columns in tables.
* Index: a table or other data structure used to determine in a file the location of records that satisfy some condition
* Primary keys are automatically indexed
* Other fields or combinations of fields can also be indexed; these are called secondary keys (or non unique keys)

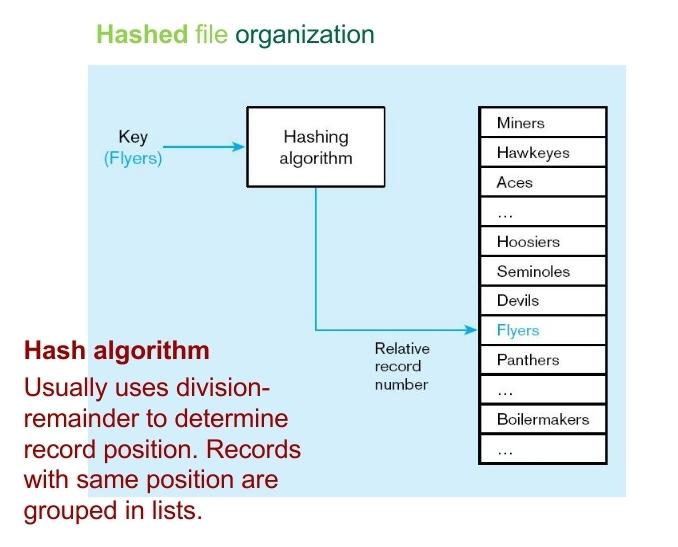


* This divides the data in your database
* This way it can scan through a limited range of data

Indexed file organization



Hashed file organization



* A hash is a fingerprint of the data

Unique and non unique indexes

* Unique (primary) Index
  + Typically done for primary keys, but could also apply to other unique fields
* Nonunique (secondary) index
  + Done for fields that are often used to group individual entities (e.g. zip code, product category)



* (Try not to use use \_T)

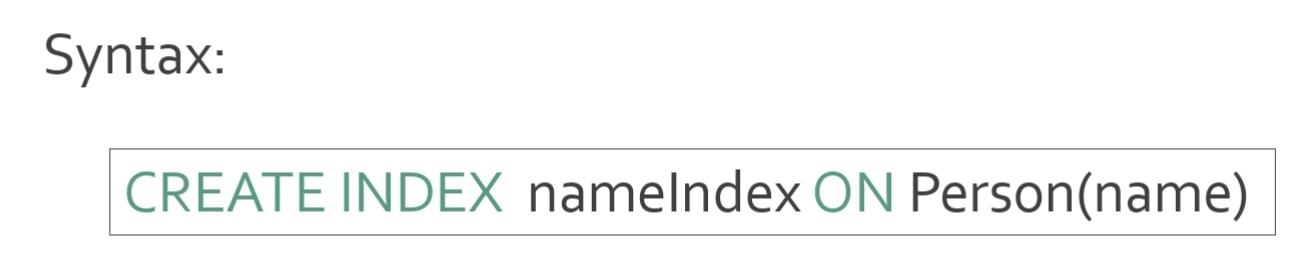
Rules for using indexes

1. Use on larger tables
2. Index the primary key of each table
3. Index search fields (fields frequently in WHERE clause)
4. Fields in SQL ORDER BY and GROUP BY commands
5. When there are >100 values but not when there are <30 values
6. Avoid use of indexes for fields with long values; perhaps compress values first

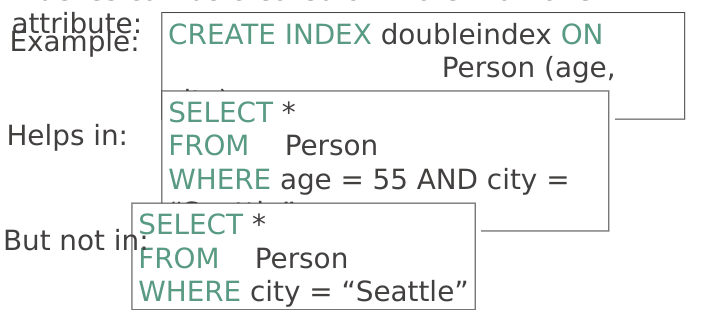
* Use content words instead of function words

1. If key to index is used to determine location of record, use surrogate (like sequence number) to allow even spread in storage area
2. DBMS may have limit on number of indexes per table and number of bytes per indexed field(s)
3. Be careful of indexing attributes with null values; many DBMSs will not recognize null values in an index search

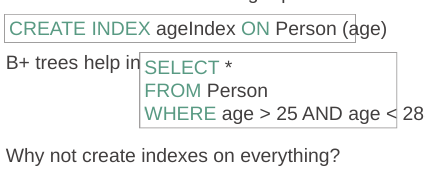
Creating Indexes



* Indexes can be created on more than one attribute:



* Indexes can be useful in range queries too:

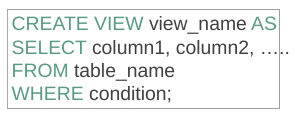


* Space - An index takes up 80% of the bites for the data being indexed
* Can cause the query operator to be confused

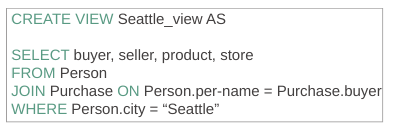
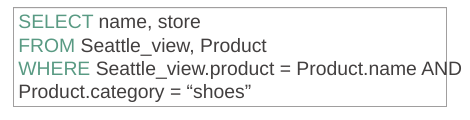
Defining Views

* Think of this as a bookmark
* Views are relations, except that they are not physically
* stored.
* For presenting different information to different users
* Employee(ssn, name, department, project, salary)
* 
* Payroll has access to Employee, others only to Developers. This means that we can control what developers see in the view and confidential information is protected.

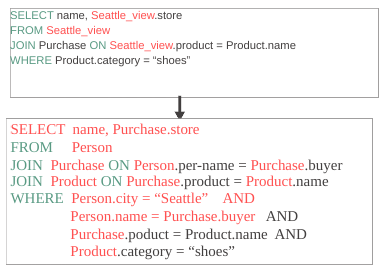
Create View in MySQL

* Note that views always contain up to date information. They are loaded at run time every time the query is run.
* 
* Dynamic because they are constantly being reloaded

A Different View

* Person(name, city)
* Purchase(buyer, seller, product, store)
* Product(name, maker, category)
* 
* We have a new virtual table:
* Seattle-view(buyer, seller, product, store)
* We can later use the view:
* 

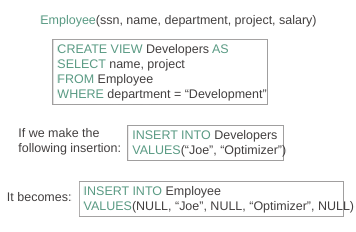
What Happens When We Query a View?

* A view is not a true table, it does not contain data
* When we query a view, it accesses the source table to capture the data
* This means that if data needs to be modified, it must be modified at the table level
* 

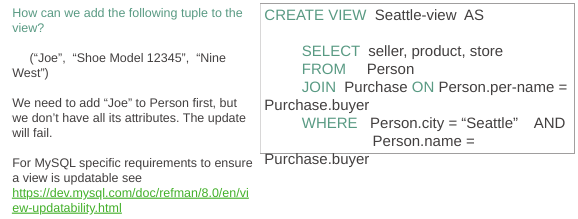
Types of Views

* Virtual views:
  + Used in databases
  + Computed only on-demand – slower at runtime
  + Always up to date
* Materialized views
  + Used in data warehouses
  + Precomputed offline – faster at runtime
  + May have stale data

Updating Views

* How can I insert a tuple into a table that doesn’t exist?
* 
* Many methods to update views.
* The first below will replace the view with new changes in the query
* The second will update a value in the source table

Non-Updatable Views

* 

Dropping Views in MySQL

* To prevent errors by dropping views that don’t exist, use the IF EXISTS clause.
* 

Materialised views

* Often used in data warehousing
* Allows for the dashboard to load instantly

You have a lab due February 18th! Dan prefers a text file (.txt) please :)