CS 143(Database Systems) Notes

Lecture Notes

10/2 - Week 1

- The Relational Data Model is the most significant invention in the history of DBMS (database management system).
    - Simplicity at the logical level because a system optimizer takes care of performance.
- Simplicity at the logical level: a system optimizer takes care of performance.
- In the relational model, all data is represented in tuples, and grouped into relations.
    - All data is represented as relations (= tables)
    - Each relation has a set of attributes (= columns)
    - Each relation contains a set of tuples (= rows)
    - Each attribute has a domain (= type)
- Can't have sets of items inside of each cell.
- Schema defines the tables and data types for each table. Data Definition Language (DDL)
- The query language has two flavors.
    - Relational Algebra: Used as the basis of the actual execution model.
    - Relational Calculus languages, logic oriented, and more declarative.
- A query is applied to relation instances, and result of a query is also a relation instance.
    - Schemas of input relations for a query are fixed.
    - The schema for the result of a given query is also fixed.
- SQL is called Data Manipulation Languages because besides query can be used to request the following DB changes.
    - Insert into table
    - Delete from table
    - Update records in a table.
- Relational Algebra basic operations
    - Selection - Selects a subset of **rows** from relation, based on some condition.
    - Projection - Deletes unwanted **columns** from relation and removes duplicates from those columns
    - Cross product - Allows us to combine two relations. Each row of S1 is paired with each row of S2.
    - Set difference - Tuples in relation 1, but not in relation 2.
    - Union - Tuples in relation 1 and in relation 2.
        - Need to have the same columns
- Each of the above operations returns a relation, and thus the operations can be composed.

- An equi-join is similar to a cross product except only one copy of fields for which equality is specified.
    - A natural join is an equijoin on all common fields.
    - There are also general joins?
- In let calls, the binding of variables is just in that scope.
- The lack of the star in let means that the expressions will be evaluated in parallel.

10/4 - Week 1

- Query statements in SQL start with the keyword select, and return a result in table form.

```
select    Attribute … Attribute
from      Table … Table
[where    Condition]
```

- The 3 parts are usually called target list, from clause, and where clause.
- Relationship between SQL and Relational Algebra
    - Cartesian product = from
    - Selection = where
    - Projection = select
- To sort a column, use order by
- You can alias by doing “select income/4 as quarterlyIncome”
- Count(*) counts the number of rows in the result.
- Sum, avg, max, and min are aggregate operators where the argument can be an attribute or an expression but can’t be *.
- You can group by a particular column, and see statistics (like the above) for each group.

SQLSelect ::=

```
select    ListOfAttributesOrExpressions
from      ListOfTables
[ where   ConditionsOnTuples ]
[ group by ListOfGroupingAttributes ]
[ having  ConditionsOnAggregates ]
[ order by ListOfOrderingAttributes ]
```

10/6 - Week 1

- Dynamic language used to create websites
- <?php code here ?>
- Variables in PHP start with a $ symbol
    - Don’t need to be declared
    - Mostly like Python

- Concatenation operator
    - <?php
      $txt1 = “Hello”;
      $txt2 = “World”;
      echo $txt1.” ”.$txt2
      >
- For the project, need to create the 5-6 tables, explicitly write down constraints of data integrity, and create the PHP web app to display the results.

10/9 - Week 2

- In SQL, you can replicate Union with the keyword union and you can replicate set difference with the keyword except.
    - With union, duplicates are eliminated, but with “union all” duplicates are kept.
    - SQL renames the attribute names of the secon
- For the union operation, SQL will rename the attributes of the second operand.
- Any expression with the operator exists is true if the result of the subquery is not empty.
- A left join keeps information from the left table of the join even if there is no corresponding information about the right table.
- Full outer join will do both a left join and a right join.
- If you just do a normal join, then unmatched values from the left and the right are lost. If you want that functionality, you need to specify left outer join or right outer join or full outer join.
- The join operations that don’t preserve non matching tuples are inner join operations.

10/11 - Week 2

- Foreign key is like a pointer.
- Types of key constraints:
    - NOT NULL - The value for that attribute can’t be null
    - UNIQUE - Can’t be duplicates but there can be null
    - PRIMARY KEY - There can only be one, can’t be null, and must be unique
    - FOREIGN KEY - Field that is primary key in another table
    - CHECK (<condition>) - Checked whenever a tuple is inserted/updated
- You can also do create assertion <assertion name> CHECK <condition> if you want. This means that you don’t necessarily have to know your checks right when you create the table. You can add it in later.
- More on triggers

Trigger: Event-Condition-Action rule

```
CREATE TRIGGER <name>
<event>
   <referencing clause>   // optional
WHEN (<condition>)        // optional
<action>
```

- <event>
  - BEFORE | AFTER INSERT ON R
  - BEFORE | AFTER DELETE ON R
  - BEFORE | AFTER UPDATE [OF A1, A2, ..., An] ON R
- <action>
  - Any SQL statement
  - Multiple statements should be enclosed with BEGIN ... END and be separated by ;
- <referencing clause>
  - REFERENCING OLD|NEW TABLE|ROW AS <var>, ...
    - FOR EACH ROW: row-level trigger
    - FOR EACH STATEMENT (default): statement-level trigger

10/13 - Week 2

- SQL (Structured Query Language)
  - 2 sublanguages
    - DDL - Data Definition Language: Define and modify schema. This is where we create the tables.
    - DML - Data Manipulation Language: Query writing
- Evaluation order in SQL
  - FROM -> WHERE -> GROUPBY -> HAVING -> ORDER BY -> SELECT
- When you have multiple tables in the FROM clause, then it will do a cross product between them. It does not do a natural join. You could make it an equijoin with certain clauses.
- Evaluation in SQL
  - Do FROM clause: Compute cross product of tables
  - Do WHERE clause: heck conditions, and discard tuples that fail.
  - Do SELECT clause: Delete unwanted fields.
  - If DISTINCT is applied, eliminate duplicates
- A primary key allows you to uniquely determine the other info in the table. For example if you're given a UID, you can identify every other piece of student information.
  - Can't be NULL and can't have any duplicates.
  - A table can only have one primary key, but that key can have multiple fields.
- When using intersect, union, and except, the set operators should have the same schema for the operands.
  - In practice, it is okay to just have compatible types.
- Drop table student

    - Deletes the table and its contents
- Alter table r add A D
    - A is the name of the attribute, r is the relation, and D is the domain of A
    - All tuples are assigned null as the value for the attribute
- Alter table r drop A
    - A is the name of an attribute of relation r.
- Delete from Instructor where dept=”CS”
    - Deletes all instructors in CS
- If you have something like this “Delete from instructor where salary < (select avg(salary) from instructor)”, the average may change as you delete tuples, so in SQL you first compute the average, find the tuples to delete and delete them without recomputing an average.
- String operations
    - Percent: The % character matches any substring
    - Underscore: The _ character matches any character.
- Unique construct tests whether a subquery has any duplicate values in its result.

10/16 - Week 3

- Data cube is a multi dimensional generalization of a cross tab.
- CUBE generates a result set that shows aggregates for all combinations of values in the selected columns.
- ROLLUP generates a result set that shows aggregates for a hierarchy of values in the selected columns.
- JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity) are two APIs for a program to interact with a database server. These are created so that you can access databases from projects written in other languages.
- APIs make calls to
    - Connect with the server
    - Send SQL commands
    - Fetch tuples
- JDBC model for communicating with database.
    - Open a connection
    - Create a statement object
    - Execute queries using that object to send queries and fetch results.
    - Exception mechanism to handle errors.
- You can also have embedded SQL which work in languages like C, Java, and Cobol.
- Language to which SQL queries are embedded is referred to as the host language, and the SQL structures permitted in the host language comprise embedded SQL.
- From within the language, you have to write EXEC SQL (command) END_EXEC
- Can use cursors for updates

Updates Through Cursors

- Can update tuples fetched by cursor by declaring that the cursor is for update

```
declare c cursor for
  select *
  from instructor
  where dept_name = 'Music'
for update
```

- To update tuple at the current location of cursor *c*

```
update instructor
set salary = salary + 100
where current of c
```

- Trigger is a statement that is executed automatically by the system as a side effect of a modification to the database. We need to
  - Specify the conditions under which the trigger is to be executed.
  - Specify the actions to be taken when the trigger executes.
- Triggers are a good view to enforce integrity constraints.

```
create trigger timeslot_check1 after insert on section
referencing new row as nrow
for each row
when (nrow.time_slot_id not in (
        select time_slot_id
        from time_slot)) /* time_slot_id not present in time_slot */
begin
   rollback
end;
```

- Triggers on event can be insert, delete, or update and triggers on update can be restricted to specific attributes.
- Triggers were used earlier for maintaining summary data and replicating databases, but now we have built in materialized view facilities to maintain summary data.
- Recursive queries can deal with graphs and trees effectively.
- Recursive view required to be monotonic, they can't be defined using difference or set aggregates in recursive rules.
- Data that can modeled as dimension attributes and measure attributes are called multidimensional data.
  - Measure attributes measure some value
    - Attribute number of the sales relation
  - Dimension attributes define the dimensions on which measure attributes are viewed.
    - Item_name or color of the sale relation

- Cube operation computes union of group by's on every subset of the specified attributes.

  select *item_name, color, size,* **sum**(*number*)
  from *sales*
  group by cube(*item_name, color, size*)

  This computes the union of eight different groupings of the *sales* relation:

  { (*item_name, color, size*), (*item_name, color*),
  (*item_name, size*), (*color, size*),
  (*item_name*), (*color*),
  (*size*), ( ) }

- Indexes are data structures used to speed up access to records with specified values for index attributes.
  - Create index studentID_index on student (ID)
- Now that we have that index, the query "select * from student where ID= 123" won't take take as long because we won't have to look through every value in student.

10/23 - Week 4

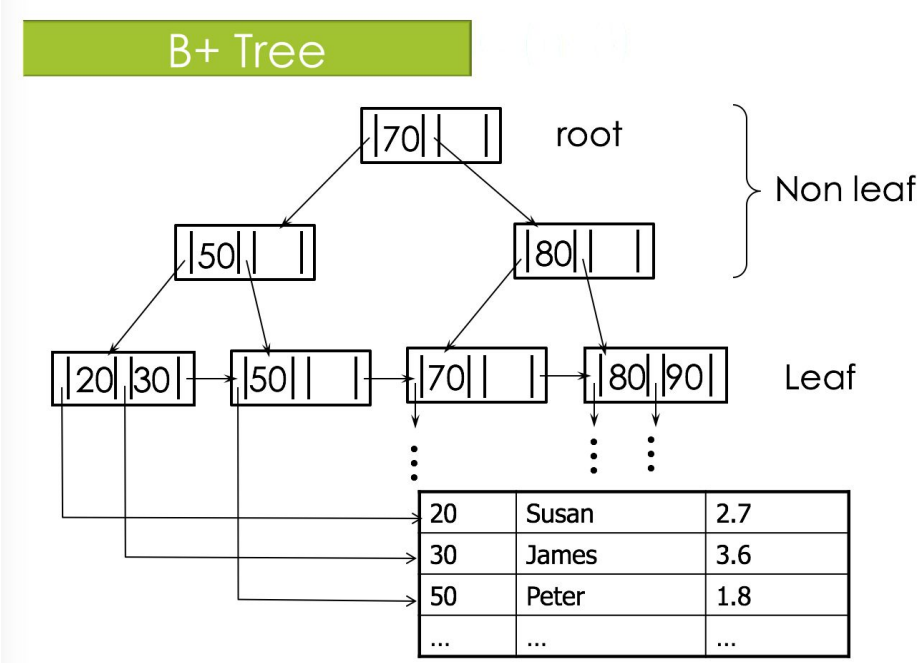
- Magnetic disk stores data on a magnetic disk while solid state drive stores data in NAND flash memory. It's faster and more reliable that magnetic disk but it is 10x more expensive.
- A typical disk has 2 platters, 2 surfaces per platter, 5000 tracks per surface, 1000 sectors per track, 1 KB per sector, and 6000 RPM.
- The overall capacity is the first 5 numbers multiplied together.
- Access time = seek time + rotational delay + transfer time
  - Seek time is the time to find the target track
    - Typical average is 10 ms
  - Rotational delay is the time to rotate the target sector
    - For 6000 RPM, it is 0.5*(1/6000) = 5ms
  - Transfer time is the time to read one block
    - Read a track, rotate a circle = 1 min/6000
    - Read one sector(block) = 10 ms/track / 1000 sector/track = 0.01 ms/sector
    - Transfer rate: 1 KB / 0.01 ms/sector = 100 MB/s
- For sequential access (aka 3 blocks in a row), you'll get 10 + 5 + 3(0.01) because seek and rotational time won't affect it because we're in the same location. However, when you have random access, the total time is 3(10 + 5 + 0.01). Thus avoid random I/O as much as possible.
- RAID refers to Redundant Array of Independent Disks
  - Create a large capacity disk volumes from an array of many disks.
  - Has potentially high throughput as you can read from multiple disks concurrently, but has potential reliability issues where one disk failure may lead to the entire disk volume failure.

- Spanned tuples are where the data blocks are squished together while unspanned is where they take up the whole row even if they don't fill it. Similar to reserved space and variable length space.
- Slotted page is where we have a bunch of pointers to places in memory.
- For long tuples, spanning or splitting the tuples is the best course of action.
- Query by example (QBE) is another relational language.

10/27 - Week 4

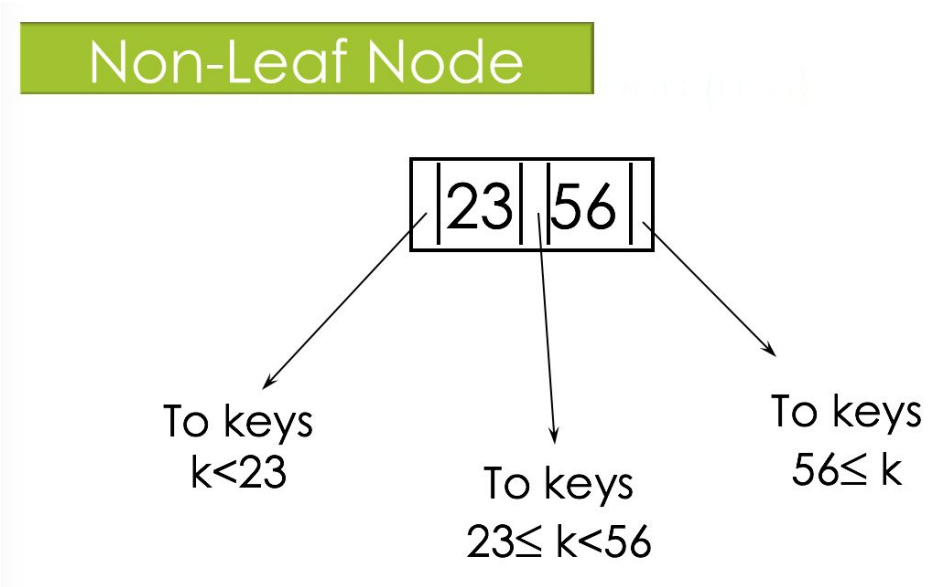
- Privileges for users for a particular relation
    - Select, Insert, Delete, Update, Grant (giving privilege to other people), Revoke
        - Revoke has several options (Cascade and Restrict) where if you revoke for one user, it could impact others.
        - Cascade means that any grants made by the person who got their privileges revoked are not enforced anymore.
        - Restrict means that if privilege has been passed to others, the revoke fails as a warning that something else must chase it down.
    - Select/Read - Right to query the relation
    - Insert - Right to insert tuples
    - Delete - Right to delete tuples
    - Update - Right to update tuples
    - Index - Right to create and delete indices
    - Resources - Right to create new relations
    - Alteration - Right to add or delete attributes in a relation
    - Drop - Right to delete relations
- Grant <privilege list> on <relation name> to <user list>
    - User list can be a user-id or public or a role
- Revoke <privilege list> on <relation name> to <user list>
- Granting a privilege on a view does not imply granting any privileges on underlying relations.
- Privileges can be granted to roles
    - Create role instructor;
    - Grant instructor to Amit;
    - Grant select on takes to instructor
- Roles can be granted to users and to other roles.
    - Grant TA to instructor
- Users are referred to by an authorization ID
- You have all possible privileges on objects that you create.
- You can grant privileges to other users and you can grant WITH GRANT option so that they can grant as well.
- Tree Structured Indexes work to speed up searches on a particular key. They support both range searches and equality searches.

    - ISAM (Indexed Sequential Access Method) is a static structure and was an early index technology. Not good when there are frequent updates.
        - Advantages are that it is simple and has sequential blocks
        - Disadvantages are that it is not suitable for updates and becomes ugly and loses balance over time.
    - B+ tree is dynamic adjusts under inserts and deletes.
        - Advantages are that it is suitable for dynamic updates, is balanced, and has a minimum space guarantee
        - Disadvantages are that you have non-sequential index blocks.
- The idea behind indices is that we have one index file that points to different pages within the actual data file.
- ISAM will index entries with <search key value, page id> and each node can hold 2 entities.
- The problem with ISAM is the this tree approach with overflow pages is that the tree quickly loses balance and becomes non sequential due to the overflow chain.
- B+ Tree consists of leaf nodes and non-leaf nodes
    - N pointers and (n-1) keys per node
    - Keys are sorted within a node
    - All leaf nodes are at the same level.



Balanced: All leaf nodes are at the same level

- A non-leaf node points to the nodes one level below.



- A non leaf node with k children has k-1 keys. This is shown above because the node has 2 keys and has 3 children.
- To maintain a balanced tree, all leaf nodes appear in the same level.

10/30 - Week 5

- For finding rows with specific values for their attributes, you'd have to do a O(N) search to get to that value is you have everything randomly placed.
- The solution is to build an index on the table, which is a structure that helps us locate a record given a key.
- You have a primary index which is an index on the search key, and you also have a dense index which is a (key, pointer) pair for every record.

11/1 - Week 5

- Problem: Blocks can hold 100 search keys and 101 pointers. Dense index on 2 million records, where records are placed in blocks that hold 10 records each. Search key is the candidate key of the relation.
    - Compute the number of blocks used at each level of the B+ tree.
    - Candidate key: As many pointers are there are records. 2 million pointers
    - Bottom level: 2 million / 50 = 40,000
    - Next level: 40,000 / 51 = 784 (Take the floor because blocks need to be at least half full)
    - Next level: 784 / 51 = 15
    - The root
    - Total: 40,000 + 784 + 15 + 1
    - If it were sparse index, then we need 200,000 pointers, then 4,000, then 78, then root.

- Problem: We have a block with size 4096 bytes. Each tuple is 44 bytes. Relation has 2 million tuples stored unspanned. Create a sparse index on id organized as a B+ tree. Pointers take 10 bytes.
    - How many blocks needed to store the relation?
        - 4096/11 = 93.09 so we can store 93 unspanned tuples in each block. Then 2000000 / 93 = 21,506 blocks
    - Minimum number of nodes needed for the B+ tree?
        - Best case: N - 1 = [(4096 - 10) / (20 + 10)] = 137 = N
        - Next level will have 136 pointers. Sparse index pointing at 21,506 blocks. So 21506 / 136 = 158.13 so 159 leaf nodes used.
        - First level: 159/137 so 2 blocks
        - 1 root
        - Worst case: Still have N = 137. Then [137 + 1 / 2] = 69/ Thus 68 pointers

11/3 - Week 5

- In extendible hashing, if you increase the local depth and local depth > global depth at some point, then we want to double the directory size and we'd need to copy pointers.
- X -> Y means that Y is functionally dependent on X. This means that
    - For every tuple t1 and t2, if t1[X] = t2[X], then t1[Y] = t2[Y]
    - No two tuples in the schema can have the same X with different Y values.
- Trivial FD is X -> Y where Y is a subset of X.

11/8 - Week 6

- Entity is an object that is distinguishable from others.
- These entities have attributes
    - Single valued attributes
    - Multi valued attributes
    - Derived attributes
- Composite attributes are made of other attributes
    - Name: First, Last, MI
- A super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A candidate key is a minimal key??
- Degree of a relationship refers to number of entity sets that participate in a relationship set.
    - 2 entity sets = Binary
- https://www.dlsweb.rmit.edu.au/toolbox/Database/Certificate4_DB_Toolbox/content/keys/keys.htm

11/17 - Week 7

- A => BD if and only if A => B and A => D

- Database normalization is the process of organizing the columns and tables of a database to reduce data redundancy and improve data integrity.
- Problems with unnormalized relations
    - Redundancy
    - Update anomaly
    - Insertion anomaly
    - Deletion anomaly
- 1NF are when you have flat tables with no structured fields.

11/20 - Week 8

- Transaction is a sequence of SQL statements executed at once.
- Main reasons for wanting transactions
    - Handle crashes (classic example of transferring money)
        - To handle these, if the system crashes after S1 then we look at our log of actions and see if S2 hasn't been executed, then we go back and undo S1.
    - Allowing concurrent execution of multiple transactions
        - So that we can have concurrent access from multiple clients
- Transactions should be ACID
    - Atomicity: All or none of the operations executed
    - Consistency: Database is in a consistent state before and after the transaction.
    - Isolation: Even if multiple transactions are executed concurrently, the result is the same as executing them in a sequential order.
    - Durability: If a transaction is committed, all changes remain permanent even after a system crash.
- During crash recovery, there are differences between setting and writing the values for variables.
    - If crashes occur after writes you're good, but if they happen after sets, that's when problems will pop up.
- A transaction that has completed all its reads and writes will commit.
- In SQL, in the absence of begin/end transaction statements, each SQL update statement is treated as a separate transaction.
    - A transaction finishes when commit or rollback statement is executed.
- Concurrent transactions allows for increased processor and disk utilization and reduced average response time for transactions.
- Concurrency can have problems with the operations are not commutative.
- Recoverable schedule — if a transaction Tj reads a data items written by a transaction Ti, then Ti must commit before Tj can commit.
- Cascadeless schedule - if a transaction Tj reads a data item previously written by Ti, then then the commit operation of Ti appears before the read of Tj. Every cascadeless schedule is also recoverable.

- A serial schedule is one where one transaction follows another. You don't have some tasks of Transaction A get executed and then some of Transaction B.
- A schedule is conflict serializable if you can swap non conflicting instructions and get a serial schedule out of it.
- Two operations are conflicting if they occur in different transactions, if they operate on the same object, and if at least one of them is a write operation.

11/27 - Week 9

- Different types of failures
    - Transaction errors
        - Logical errors - Could be violation of integrity constraint
        - System errors - Could be deadlock
    - System crash - Could be power failure
        - Non-volatile storage is assumed not to be corrupted.
    - Disk failure - A head crash destroys all or part of the disk storage.
        - Disk drives use checksums to check for this type of failure.
- Recovery needs to make sure that
    - Actions taken during the transaction needs to ensure that enough information exists such as in the form of a log.
    - Actions taken after a failure ensures atomicity, consistency, and durability.
- Storage types
    - Volatile storage doesn't survive system crashes
        - Main memory and cache memory
    - Nonvolatile storage survives system crashes
        - Non volatile RAM, flash memory
    - Stable storage
        - Maintain multiple copies on distinct non volatile media. RAID is an example of this.
- Physical blocks are those that reside on disk while buffer blocks reside temporarily in main memory.
    - Input(B) transfers the buffer block B to main memory.
    - Output(B) transfers the buffer block B to __?
- Log based recovery mechanisms are used to ensure atomicity despite the failures.
- Log descriptions
    - <T, start> represents the start of a transaction log.
    - <T, X, V1, V2> shows the log when write(X) is executed. V1 is the value before the write and V2 is the value to be written to X.
- Undo of a log record will write the old value V1 to X.
    - A transaction needs to be undone if the log contains <T, start> and doesn't contain either <T, commit> or <T, abort>
- Redo of a log record will write the new value V2 to X.
    - Same condition???

- Immediate modification scheme is where you write to the log before the database items are actually affected.
- Deferred modification scheme is when it writes to log when the commit point is reached.
- A transaction is said to have committed when its commit log record is output to stable storage.
    - All previous log records of the transaction should have been outputted already,

11/29 - Week 9

- Distributed transactions may access data at several sites.
- Each site has a transaction manager and transaction coordinator.
- Commit protocol used to ensure atomicity across sites.

12/4 - Week 10

- Foreign keys for a particular table can be a relationship between that table and another table.
- If you have a primary key for a table, then it can uniquely determine the rest of the attributes. If you have (eid, did) with eid being the primary key, you know that an employee can only work in one department.
- When you see a REFERENCE in a schema and you have to draw an ER diagram, that normally means that you should specify a relationship between two entities.
- When you're trying to give a lossless decomposition
    - Separate into the two relations. For each relation
        - Look at the associated FDs for those entities and do the check that for every nontrivial FD, X must be a key of the relation (aka its closure has to be that whole set).
        - To decompose a relation another time you should use an FD that failed the BCNF check
- TODO figure out how to draw conflict serializability graphs
- Topics for Final
    - Given a schema, be able to draw an ER diagram

12/8 - Week 10

- Final Review topics
    - ER Diagrams
    - Functional Dependencies and Normal Form
    - Transactions
- When you have an entity set where even all the attributes put together cannot be unique or something and has to rely on another entity set, then it is a weak one.
- K is a superkey for schema R if and only if K -> R

- Recoverable schedule - If a transaction reads a data item previously written by another transaction then the commit operation of that 2nd operation must appear before the commit of the first.

Textbook Notes

Chapter 1 - Introduction

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data.
    - Collection of data is called the database.
    - Provides users with an abstract view of the data.
- Database systems have to define structures for storage of info as well as provide mechanisms for the manipulation of that information.
- Some issues you have to deal with when storing data
    - Data redundancy and inconsistency if you're working with a lot of files of data.
    - Difficulty in accessing data
    - Data isolation
    - Atomicity problems
    - Concurrent access
- We have data abstractions at the physical, logical, and view level.
    - Physical level: Describes how the data is actually stored.
    - Logical level: Describes what data are stored in the database and what relationships there are.
    - View level: Simplify interaction with the system.
- The collection of information stored in the database at a particular moment is called an instance of the database.
- Overall design of the database is called the database schema.
- Data models are tools for describing data and data relationships.
    - Relational model - Collection of tables to represent data and relationships. Tables are known as relations
    - Entity relationship model - Collection of basic objects, where those entities have sets of attributes, and relationships among those objects.

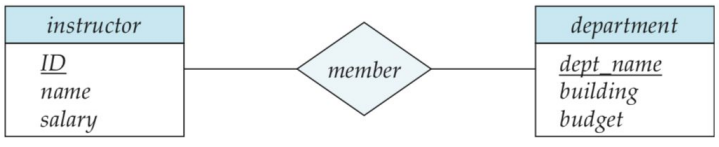


Figure 1.3 A sample E-R diagram.

    - Object based data model - Extension of ER model with encapsulation and methods and object identity.
    - Semistructured data model - Individual data items of same type can have different attributes.

- Data manipulation language allows user to access/manipulate data.
    - Procedural DMLs require user to specify what data are needed, how to get that data
    - Declarative DMLs require user to specify what data are needed, without specifying how to get it.
- Data definition languages specify database schemas.

Chapter 4 - Intermediate SQL

- When you're trying to do a join, but you don't want values to be lost if the join category in one example is not there in the another table. This would normally mean that that example would be left out.
    - Outer join preserves the tuples that would be lost in a join.
    - Left outer join preserves tuples only in the relation named to the left of the left outer join operation.
    - Right outer join preserves tuples only in the relation named to the right of the right outer join operation.
    - Full outer join does both.
- Virtual relations are used to give users a view of the relations in the table that they are allowed to see. They can't see the full table, but they can see certain attributes or certain examples. A relation that is not part of the logical model but is made visible to a user as a virtual relation is called a view.

```
create view faculty as
select ID, name, dept_name
from instructor;
```

- As shown above, that particular view will only show those categories from the main instructor table.
- Basically, SQL will store that query expression so that it can use it later on, but doesn't have to actually compute the results at the moment.
- Then, you can use view as the argument to FROM
- Materialized views are where the results from the relation specified are actually stored, but if there is a change to the original table, then the view is kept up to date.
    - Process of keeping the view up to date is called materialized view maintenance.
- Trying to insert something into a view may result in an error.
- SQL view is updatable if
    - From clause has one database relation
    - Select clause contains attribute names of the relation (no expressions, aggregates, or distinct specification)
    - Any attribute not listed in select can be set to null
    - No group by or having clause

- A transaction consists of a sequence of query and/or update statements. Transaction begins implicitly when a SQL statement is executed. A commit work or rollback work statement must end the interaction.
- If we want to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation, this is called referential integrity.
    - Foreign key dept references department means that for each course tuple, the department name specified by dept must exist in the department relation.

Chapter 5 - Advanced SQL

- JDBC standard defines an API that Java programs can use to connect to database servers.
- ODBC standard defines that API that applications can use to open a connection with a database, send queries, and get back results.
- Difference between dynamic SQL and embedded SQL is that in dynamic, you have a program that can construct and submit queries at runtime. In embedded SQL, the SQL statements are identified at compile time using a preprocessor. It submits the SQL statements to the database for precompilation and optimization, and then replaces the code inside the program with the appropriate function calls.
- A language in which SQL queries are embedded is referred to as a host language.
- Operation of moving from finer granularity data to a coarser granularity by means of aggregation is called a rollup.
    - The opposite is called a drill down.

Chapter 6 - Formal Relational Query Languages

- Relational algebra forms the basis of the SQL query language.
- Relational algebra is a procedural query language consisting of a set of operations that take in relations as input and produce a new relation as output.
- Select, project, and rename are unary operations.
    - Select selects the tuples which satisfy a given predicate.
        - The lowercase sigma denotes selection.
    - Project returns its argument relation, with certain attributes left out.
        - Uppercase pi denotes projection
    - Rename allows you to give names to the results of relational algebra.
        - Lowercase rho denotes renaming.
- Union, set intersection, set difference, and Cartesian product are binary operations.
    - Union takes the data that appears in either set, and set intersection takes the data that appears in both sets.
        - Duplicate values are replaced by a single occurrence.
        - Must ensure that unions are taken between compatible relations.

        - For the union to be valid, the relations r and s must have the same number of attributes and the domains of the ith attribute of r and the ith attribute of s have to be the same.
    - Set difference lets us find tuples in one relation but not in another.
        - Same rules for compatible relations.
    - Cartesian product lets us combine information from any two relations.
- A natural join is a binary operation that allows us to combine certain selections and a Cartesian product into one operation.
    - Forms a Cartesian product of its two arguments, performs a selection forcing equality on those attributes that appear in both relation schemas, and removes duplicate attributes.

Chapter 7 - Database Design and the E-R Model

- Entity relationship data model provides a means of identifying entities to be represented in the database and how those entities are related.
- Entity is any distinctly identifiable item.
    - Represented by a set of attributes, and each entity has a value for each of its attributes. Some of these attributes can be unique identifiers.
- In a database, want to avoid redundancy (repeated info) -> data may become inconsistent, and want to avoid incompleteness.
- A relationship is an association among several entities.
    - The function that the entity plays in the relationship is its role.
- E-R data model has 3 basic entities.
    - Entity sets - Set of entities that share the same properties, or attributes.
        - Don't need to be disjoint.
    - Relationship sets - Set of relationships of the same type
        - Examples is a takes relationship set that considers the entity sets student and section.
        - Entity sets are said to participate in the relationship set R.
        - Can also have descriptive attributes.
    - Attributes
        - The set of permitted values for an attribute is the domain.
- A binary relationship set involves 2 entity sets.
- Number of entity sets that participate in a relationship set is the degree of the relationship set.
- Composite attributes can be divided into other attributes.
    - Name could be firstName, middleName, lastName
- Attributes can be single valued or multivalued.
    - For the multivalued ones, just enclose all the terms in braces.
- Derived attributes are those where you can figure out the value of the attribute based on characteristics of the other attributes.

    - Examples are that you can get the num_students_advised by counting the items in student.
- Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set.
    - For a binary relationship between A and B, the mapping must be one to one, one to many, many to one, or many to many.
- The participation of entity set E in the relationship set R is total if every entity in E participates in one relationship in R.
    - Participation is partial if there are some that aren't in a relationship.
- No two entities in an entity set are allowed to have exactly the same value for all attributes.
- A key for an entity is a set of attributes that suffice to distinguish entities from each other.
    - Primary key of an entity allows us to distinguish among various entities of the set.
- A superkey is the union of all the primary keys of all the entities in a relationship set.
    - Candidate key is the minimal super key. It means that if you remove one of the attributes then you won't have a super key left.
        - In other words, it's the minimum set of attributes that can uniquely identify a tuple. A superkey is any set that uniquely identifies a tuple.
- If an attribute appears in both entity sets A and B, and the two sets have some relationship R and the attribute is a primary key in one of the entity sets, then you should remove the attribute from the other set since it is redundant.
- If you have a non binary relationship set,then you can have at most one arrow pointing out.
- An entity set without sufficient attributes to form a primary key is termed a weak entity set.
    - In other words, since there is no primary key and thus you can uniquely identify the members just based on one or more attributes, we have to use a type of foreign key.
        - It is typically a primary key that the entity is related to.
    - For it to be useful, it has to be associated with another entity set called the owner entity set.
- An entity set that has a primary key is a strong entity set.
- Discriminator of a weak entity set is a set of attributes that allows the distinguishing of a particular entity.
    - Normally, if you have a weak set, I feel like it would likely just be all of the attributes.
    - Discriminator is also called the partial key of the entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.
- Relationship set that connects the weak entity set to the identifying strong entity set is shown by a double diamond.
    - The double lines show total participation. The weak set will likely have this.

- You can also have a weak entity set that has multiple identifying entity sets.
- When you have multivalued attributes, you can only represent them in your table if you have a separate relation that maps the primary key to the attribute.
- For a binary many to many relationship, the union of the primary key attributes from the participating entity sets becomes the primary key.
- For a binary one to one relationship, the primary key attribute of either set forms the primary key.
- For many to one or one to many, the primary key of the entity set on the many side of the relationship serves as the primary key.
- For n ary relationship with no arrows, the union of the primary key attributes from the participating entity sets becomes the primary key.
    - If you do have arrows, then the primary keys of the sets not on the arrow side serve as the primary key.
- We can also create foreign key constraints on the relationship schema R where for each entity set that is related to R, we make a foreign key constraint with the attributes of R that were derived from the primary key attributes of each of the entity sets.
    - If you have a binary relation where both entities have primary keys, then that relation with have 2 foreign key constraints.
- Process of designating subgroups within an entity set is called specialization.
    - Employee and student within the entity set person.
    - Each of these has to have all the attributes that describe person and possibly some other additional attributes.
- Overlapping specialization is when entities can belong to multiple specialized entity sets.
    - Student and employee in person since someone can fit both student and employee
    - 2 separate arrows used
- Disjoint specialization is when entities can belong to at most one specialized entity set.
    - Instructor and secretary in person since someone can't have both.
    - One arrow used
- Refinement from initial entity set into successive levels of entity subgrouping shows a top down approach.
- Generalization is when there is a containment relationship that exists between a high level entity set and a lower level set.
    - Used when you have attributes that are conceptually the same, but have different names. To do generalization, they must have given a common name and represented in the entity set above the higher one.
- Attributes of the higher level entity sets are inherited by the lower level entity sets.
- Total generalization is when each higher level entity must belong to a lower level entity set.
    - Partial generalization is given some higher level entities don't belong to any lower level set.
- Aggregation is an abstraction through which relationships are treated as higher level entities.

Chapter 8 - Relational Database Design

- Functional dependency is a relationship where one attribute uniquely determines another attribute.
    - X -> Y if and only if each X value in R is associated with precisely one Y value.
    - Y is said to be functionally determined by X.
- The set of all attributes functionally determined by X under a set of FDs is the closure of X, and is shown by X+.
    - Attribute of an FD is extraneous if we can remove it without changing the closure of the set of FDs.
- Lossy decompositions are when you try to decompose a table into 2 tables, and then when you try to natural join them back together, then you loss information.
    - Lossless decompositions are when you don't loss info
- A domain is atomic if elements of the domain are considered to be indivisible units.
- A relationship schema R is first normal form or 1NF if the domains of all the attributes of R are atomic. Also, the value of each attribute should only contain a single value, no multivalues. And the rows have to be uniquely identified by some attribute.
    - Set of integers is atomic but the set of addresses (with component attributes) is not.
- https://en.wikipedia.org/wiki/Second_normal_form
    - Every attribute that is not part of the candidate key is dependent on the whole of every candidate key.
    - Go through each of those non prime keys and ask yourself whether the value of the attribute depends on the candidate key.
- https://en.wikipedia.org/wiki/Third_normal_form
    - All the attributes can be determined by only the candidate key.
    - Basically, look for two columns where there's a clear one to one relationship and if you can find that, then you know it's not in 3NF.
- An instance satisfies the functional dependency X -> Y if for all pairs of tuples t1 and t2 and t1[X] = t2[X], then t1[Y] = t2[Y]
    - The functional dependency holds on the relationship schema if for every legal instance of that schema, it satisfies the FD.
- Some functional dependencies are trivial because they are satisfied by all relations. Also, trivial FDs are where Y is a subset of X.
- A relationship schema R is in BCNF with respect to a set of FDs if for all FDs of the form X -> Y where X and Y are both subsets of R and either X -> Y is a trivial dependency or X is a superkey for schema R.
    - Basically, BCNF requires that all nontrivial dependencies have X as the superkey.
- A relationship schema is 3NF if for all FDs of the form X -> Y where X and Y are both subsets of R and either X -> Y is a trivial dependency or X is a superkey for schema R or each attribute in Y - X is contained in a candidate key for R.

- Reflexivity rule: If Y is a subset of X, then X -> Y
- Augmentation rule: If X -> Y and gamma is a set of attributes, then gammaX -> gammaY
- Transitivity rule: If X -> Y, and Y -> Z, then X -> Z
- Above 3 rules are sound and complete.

Chapter 11 - Indexing and Hashing

- To retrieve a student record given an ID, the database would look up an index to find on which disk block the corresponding record resides, and then fetch the disk block to get the appropriate student record.
- Techniques for ordered indexing and hashing include metrics on access types, access time, insertion time, deletion time, and space overhead.
- Ordered indices store both the values of the search key (what you use to look up a record) as well as the records that contain it.
- An index record consists of a search key value and pointers to one or more records with that value as their search key value.
- Two types of indices
    - Dense index - An index entry appears for every search key value in the file. The index record contains the search key value and a pointer to the first data record with that search key value.

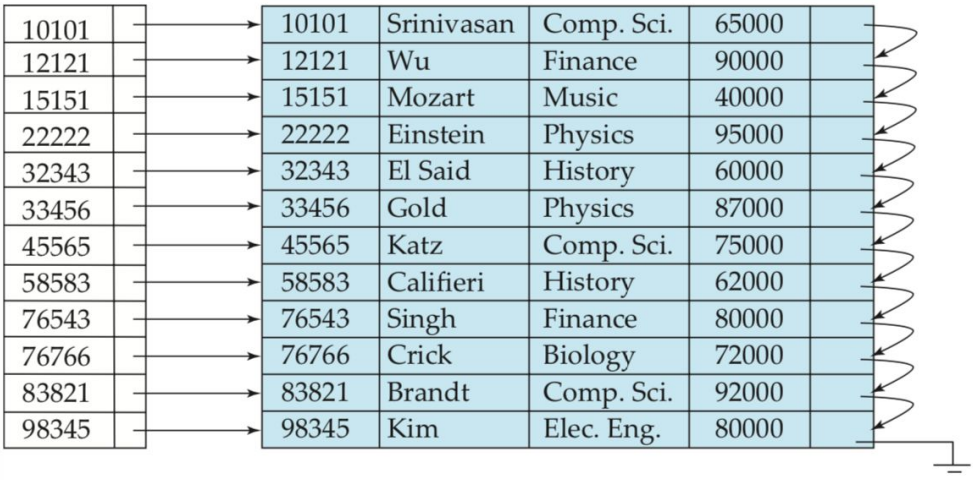


Figure 11.2 Dense index.

    - Sparse index - Index entry only appears for some of the search key values.

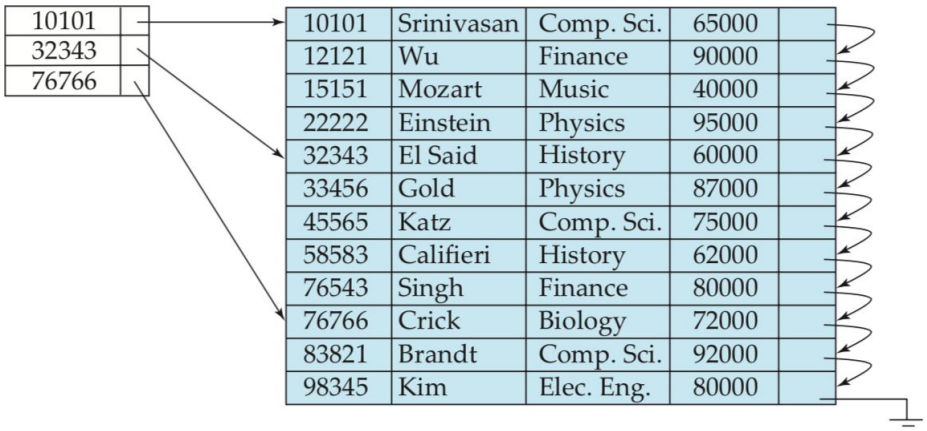


Figure 11.3 Sparse index.

- Determining what your search key is going to be is vital to the efficiency of your table.
- Multilevel indices are when you have multiple levels and inner/outer indices, etc.
- Every index must be updated whenever a record is either inserted or deleted from the file.
- For updating single level indices
    - Insertion - System does a lookup using the search-key value. For dense indices, if it's not found, the index entry is just inserted, but if it is, then it will add a pointer to whatever the first record is. For sparse, you can create a new block and insert there.
    - Deletion is similar

Primary index:

A primary index is an index on a set of fields that includes the unique primary key for the field and is guaranteed not to contain duplicates. Also Called a **Clustered index**. eg. Employee ID can be Example of it.

Secondary index:

A Secondary index is an index that is not a primary index and may have duplicates. eg. Employee name can be example of it. Because Employee name can have similar values.

- Secondary indices must be dense, with an index entry for every search key value.
- A B+ tree structure is another way to store data and it is good because it maintains efficiency despite insertion and deletion of data.
    - Balanced tree where every path from root to a leaf is the same length.
- Each node of a B+ tree has n-1 search key values and n pointers.
    - Search key values are placed in order
- Pi points to a file record with search key value Ki.
- The non leaf nodes of a B+ tree form a multilevel sparse index on the leaf nodes. All the pointers are pointers to tree nodes.
- The number of pointers in a node is called the fanout of the node.

- You can have indices on multiple keys in a relation,
    - You can also create and use an index on a composite search key (dept name, salary). You basically create an B+ tree index on the composite search key.
    - R trees can also be used for these types of queries.
- Using an index structure may also be slow sometimes because we need to locate data or do some sort of a binary search.
- We can use the technique of hashing instead. It provides a way of constructing indices through a hash function.
- The worst hash function just mpas all search key values to the same bucket, and thus to find a value, you have to search through everything in the bucket.
- We want the hash function distribution to be uniform and random.
- If a particular bucket doesn't have enough space to hold more entries, a bucket overflow is said to occur. It can happen because of insufficient buckets or because of skew.
    - We handle bucket overflow by having specific overflow buckets.
    - Overflow handling using a linked list is called overflow chaining.
- Closed hashing has no linked lists and open hashing does.
- A hash index organizes the search keys into a hash file structure.

Chapter 14 - Transactions

- Collections of operations that form a single logical unit of work are called transactions.
    - Appears to the user as a single indivisible unit.
- All or none property of transactions being executed is called atomicity.
- A transaction aborts when it doesn't complete its execution successfully.
    - Changes to the database made by the aborted transaction must be rolled back by maintaining a log.
- Dirty writes are writes to a data item that has already been written by another transaction that has not committed yet.

Chapter 15 - Concurrency Control

- Transaction can only access a data item if it has a lock on it.
- 2 phase locking ensures conflict serializability.
- Strict 2 phase locking when additional condition that all exclusive mode locks taken by a transaction be held until that transaction commits.
    - Rigorous 2 phase locking is when all locks should be held until the transaction commits.
- Tree Locking
    - When a transaction T1 performs a read of an uncommitted data item, there is a commit dependency of T1 on the transaction that performed the last write to that item, and thus is cannot be committed until the commit of all transactions on which is has a commit dependency.