**Database Design & Normalization** - Goals of database design: 1. Protect the integrity of the data. 2. Provide for change. 3. Provide access to complete data for decision making. 4. Simplify the access to data via a SQL SELECT statement. Data Anomalies: An anomaly is a potential error or inconsistency in the data (most frequently caused by M:N relationships. Insertion anomaly: Can't add "some" of a row; must have all the key attributes. Deletion anomaly: Lose some relevant data when deleting other data. Update anomaly: Must update more than one row when one piece of data changes. What is normalization?: A formal, process-oriented approach to data modeling. Normalization is the process of examining groups of data attributes, splitting them into appropriate entities, identifying the relationships between the entities, and identifying appropriate primary and foreign keys. Normalization is used to help design a database and validate a database. Functional dependency: A relationship between attributes in which one attribute or group of attributes determines the value of another. Determinant: An attribute or group of attributes that, once known, can determine the value of another attribute. Ex: SSN -> name (that's in functional dependency diagram format), address. Name and address are functionally dependent on SSN and SSN determines name and address. Unnormalized data: A data model that has not been normalized. It contains repeating groups and is not a stable model (essentially one entity). First normal form: Remove repeating groups (an attribute or group of attributes that can have more than one value for an instance of an entity, aka multi-valued attributes). Second normal form: Remove partial functional dependencies (a situation in which one or more non-key attributes are functionally dependent on part, but not all, of the primary key (occur only with concatenated keys)). Third normal form: Remove transitive dependencies (occurs when a non-key attribute is functionally dependent on one or more non-key attributes). Goal of normalization: A set of entities where each attribute is dependent on the primary key, the whole primary key, and nothing but the primary key.

**Data Management, Governance, and Administration** - Internally generated structured data: Relational DBMS, schema on write (knowing the scheme in advance), human data governance, data and database administration. Internally generated unstructured data: Individual software solutions, relational DBMS (?), schema on write (?), human data governance, data and database administration (?). Externally generated structured data: Relational DBMS, schema on write, human data governance, data and database administration. Externally generated unstructured data: NoSQL or Hadoop, schema on read (read data, figure out format, and then create the schema). Five V's of Big Data (externally generated unstructured data): Volume, variety, velocity, veracity, and value. Why do we need to govern data?: It's a critical organizational resource that is shared among multiple parts of the organization. Three key objectives of data governance: 1. Protect and control. 2. Make useful. 3. Adapt and change. Two general human-reliant components of governing data?: Data administration (responsible for the overall coordination and management of data resources) and database administration (responsible for the technical viability of the database and the DBMS). Three of the critical tasks of a DBA: 1. Database backup and recovery. 2. Database security and integrity. 3. DBMS optimization. Database backup: A method of storing data from a database in a format that can be used to rebuild the database if necessary. Database recovery: Mechanisms for restoring a database quickly and accurately after loss or damage. Transaction: One or more database actions (SQL statements) that are treated as a single unit of work. If transaction was successful, then the transaction is committed. If transaction is not successful, then the transaction is rolled back or aborted. Transaction boundary decision: Division of work into transactions. Objectives are to minimize transaction duration and ensure transaction isolation. Enforcement of important integrity constraints. Backup is conducted in three processes: 1. Backup: Can perform complete (entire database), full (all rows of specified tables), or incremental (rows that have changed since the last full backup). 2. Journalize: Provides an audit trail of changes to the database. Transaction log (contains all data used to process changes against the database) and Database change log (contains a before-image and an after-image of each row modified by a database transaction). Checkpoint: A DBMS utility that periodically suspends all transaction processing and synchronizes files within the database (writing to disk). Purpose of a checkpoint is to minimize the amount of time it takes to restore a system. Switch (recovery method): Switches to a replica of the database on a different storage device (requires a mirror image to be stored). Restore/rerun (recovery method): Reprocesses the transactions for a given time period against a correct version of the database. Backward recovery (first of two methods of restore/rerun): AKA "rollback" recovery. Used to undo unwanted changes to the database. Before-images are used to back out changes that are unwanted. Forward recovery (second of two methods of restore/rerun): AKA "rollforward" recovery. Used to recover accurate transactions and apply them to the database. After-images are applied to a past version of the DB. Database integrity: A potential problem with shared databases. Concurrency control: The process of managing concurrent operations against a DB in order to maintain data integrity (accuracy of data). Pessimistic approach (concurrency control): Assumes that every transaction could potentially be in conflict so each transaction should be controlled (older DBMS, requires a scheduler). Optimistic approach: Assumes that most transactions will not be in conflict so each transaction should be checked only when something is written to disk (newer DBMS, requires a scheduler). Locking (used in pessimistic approach): Guarantees exclusive use of a data item to a current transaction. Can be applied at a database level, table level, page level, row level, and even column level. Versioning (used in optimistic approach): the DBMS will create a new version when a transaction takes place instead of rewriting the old record. SQL statements used for data integrity: CREATE DOMAIN, CREATE ASSERTION, triggers, and stored procedures. Trigger: SQL statements that are executed when an insert, update, or delete occurs. A trigger is stored as a database object (related to a table and an event and is "fired" when the event occurs on the table). Stored procedure (batch programs): A module of code that implements business logic. They are DB objects. Unlike a trigger, stored procedures are executed when they are needed (not executed by a DB event). Database security: Protection of the data against accidental or intentional loss, destruction, or misuse. Privilege abuse (top DB security threat): Excessive access privileges, using privileges to create unauthorized data sets, enhancing privilege access from user to administrator. SQL injection (top DB security threat): Adding SQL code via web forms (or other vulnerable input channels) to execute fraudulent commands. Other top DB security threats: Weak authentication, backup data availability, and weak audit trail. Schema (helps with security): A logical collection of DB objects (tables, views, sequences, synonyms, indexes, clusters, procedures, functions, packages, and database links). SQL statements used for security: CREATE USER, GRANT, CREATE ROLE, ALTER USER, REVOKE. DBMS Query Optimization: A component of a DBMS that dictates how queries are implemented on the physical database. Query optimizer methods: Rule based (looks at syntax, parses query and executes in the order written according to the rules pre-established by the person who wrote the query optimizer), cost based (looks at syntax, looks at statistical data about DB, parses query and executes based on the written and the information about the current and historical data of the DB), and choose (uses the rule based method for tables which have not been used/analyzed in the past and uses the cost based method for tables which have been previously analyzed). Types of NoSQL DBMS: key-value stores (data is stored in rows with a key and then a value column; no detailed columns), document stores (data is stored in rows with a key and the columns are defined via the data that is input), wide-column stores (data is stored in rows with a key value and groups of columns; columns are defined via the data that is input), and graph-oriented databases (no rows and columns; uses graph structure to represent and store data). Hadoop: An open-source framework for storing data on large clusters of commodity hardware.

**Overview of Data Resource Management** - Problems with internal operational data: Not integrated. Redundant of other systems in the organization. Potentially of poor quality( “dirty data” ): Incomplete, Not accurate, Inconsistent, the meaning of the data is not fully defined and/or understood by all stakeholders. Business Intelligence “System”: Encompasses all processes, hardware and software necessary to extract data, transform it, integrate it, store it, and provide information. The information is then made effective and accessible to users to support decision making. The “V’s” of Big Data: Volume - scale of data. Velocity - frequency of change. Variety: Different forms and sources of data. Veracity: Uncertainty of the accuracy of data. Components of BIS: Data store, extraction/transformation/loading processes, end user query tools, end user visualization tools. Data Warehouse: A database designed to support a business intelligence system. Integrated: It is centralized, consolidated database integrated data from an entire organization. Subject-oriented: Data warehouse data are organized around key subjects. Time-variant: Data in the warehouse contain a time dimension so that they may be used as a historical aggregation. Non-volatile: Once data enter, they seldom leave. Data are appended rather than overwritten. Data are updated in batches. Issues in designing a data warehouse: Should have a predefined subject focus. Has the potential to be very large. Will always have a dimension of time. May contain derived data. May be a summary of data, rather than each detailed transaction. Does not always adhere to standard normalization rules. Visualization tools: Graphical, spreadsheet format, reporting tools. Query tools: OLAP: Online analytical processing. Data mining: AI based query methods. Online analytical processing: Provides multi-dimensional data analysis techniques. Works primarily with data aggregation. Provides advanced statistical analysis. Supports access to very large databases. Provides enhanced query optimization algorithms. Types of data required to be stored for an organization: Operational vs audit. Transaction vs decision support. Internal vs external. A database to store decision support data is frequently called a “data warehouse”. Issues in designing different data stores: Metadata definition, data architecture, database design, data quality, data entry, data access, information creation.

**Physical Data & Performance** - Convert Logical to Physical: Identify all necessary data attributes, Determine correct size and data type for each data attribute so that it can be a physical field, Choose an appropriate primary key, Identify foreign keys necessary to sustain relationships, Define necessary constraints. Database Performance: Minimize response time to **access** data in a database. Minimize response time to **change** contents in a database. Improving Performance: 1. By optimizing use of existing resources. 2. By using better or more resources. 3. By creating indexes. 4. By denormalizing the database. 5. By partitioning the database. 6. By storing derived data. 7. By creating procedures to archive data. Add or change resources: more memory > more processor power. Faster, more efficient disk or a fully in-memory database really helps. RAID: Redundant arrays of inexpensive disks: A set of multiple physical disk drives that appear to the designer and user as a single storage unit. Segments of data, called stripes, cut across all the disk drives. Access can occur concurrently. **Indexes:** SQL command: Create index index_name on table(column_name); File organization: The physical arrangement of a data in a file into records and pages on the secondary storage. Dictates the physical placement of records on secondary storage. File access methods: Dictates how data can be retrieved from secondary storage. Options includes: Sequential access from beginning. Sequential access from pre-defined point. Backwards from end. Backwards from pre-defined point. Direct (not really direct - has to go through a series of indices or through a hashing algorithm). File organization options: Sequential - Records are stored one after another. Hashed - The location of the records in the file is calculated based on an algorithm. Indexed - Records are stored either ordered or not in sequential organization. Additional structure, index, is built based on pre-determined keys for the records. 3 types of pointers: Physical, Relative, and Logical or “index”. Index Drawbacks: Create slower data updates and require periodic reorganization. Types of indexes: Clustered, Non-clustered, Cluster, Join. **Partitioning:** Horizontal - Split rows into separate tables. Could do it based on geographical region, date, or data types. Vertical - split columns into separate tables. Could do it based on application types( accounting, manufacturing, or inventory control). **Derived Data**: Use when aggregate values are regularly retrieved. Use when aggregate values are costly to calculate. Permit updating only of source data. Do not put derived rows in same table as table containing source.

Code syntax: SFWGHO Execution order: FWGHSO