Data Modeling:

- An Entity-Relationship Model (ERM) is a representation of structured data; entity-relationship modeling is the process of generating these models.
- The end product of the modeling process is an Entity-Relationship Diagram (ERD), a type of conceptual data model or semantic data model.
- The first stage of information system design uses these models to describe information needs or the type of information that is to be stored in a database during the requirements analysis.

Data Modeling Technique:

- This technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e., area of interest).

Entity & Entity sets:

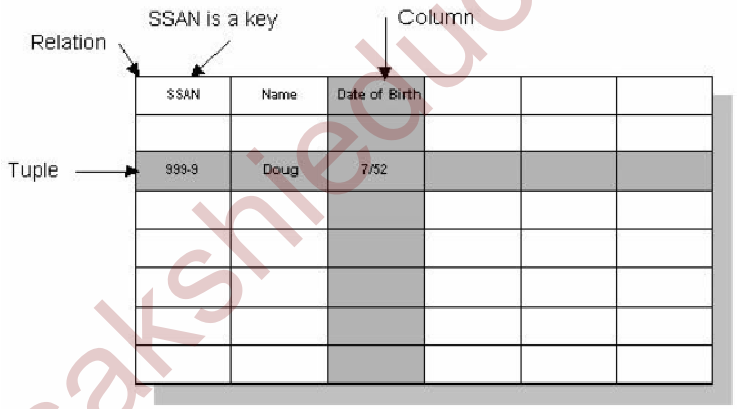
- An **Entity** is an object that exists and is distinguishable from other objects. For instance, John Harris with S.I.N. 890-12-3456 is an entity, as he can be uniquely identified as one particular person in the universe.
- An entity may be concrete (a person or a book, for example) or abstract (like a holiday or a concept).
- An entity set is a set of entities of the same type (e.g., all persons having an account at a bank).
- Entity sets need not be disjoint. For example, the entity set employee (all employees of a bank) and the entity set customer (all customers of the bank) may have members in common.

Relationship sets:

- Entity-relationship diagrams don't show single entities or single instances of relations.
- Rather, they show entity sets and relationship sets.
- Example: a particular song is an entity. The collection of all songs in a database is an entity set. The eaten relationship between a child and her lunch is a single relationship. The set of all such child-lunch relationships in a database is a relationship set.

Relational model:

- The relational model used the basic concept of a relation or table. The columns or fields in the table identify the attributes such as name, age, and so. A tuple or row contains all the data of a single instance of the table such as a person named Doug.
- In the relational model, every tuple must have a unique identification or key based on the data.
- In this figure, a social security account number (SSAN) is the key that uniquely identifies each tuple in the relation. Often, keys are used to join data from two or more relations based on matching identification.
- Relational model also includes concepts such as foreign keys, which are primary keys in one relation that are kept in another relation to allow for the joining of data.
- As an example of foreign keys is storing your mother's and father's SSAN in the tuple that represent you. Your parents' SSANs are keys for the tuples that represent them and they are foreign keys in the tuple that represents you.



Introduction to Relational Algebra and Relational Calculus:

- More precisely, relational algebra is a collection of operations that could be used to combine tables. Just as you can combine numbers using the operations of addition, subtraction, multiplication, and division, you can combine tables using operations like “selection,” “projection,” “union,” “difference,” and “join”. It documents the sequence of relational algebra operations that Oracle Database used at run-time to execute any particular query; i.e., it documents the query execution plan used by Oracle Database.

Relational Algebra Operations:

Below are fundamental operations that are "complete". That is, this set of operations alone can define any retrieval.

- Select
- Project
- Rename
- Union
- Set Difference
- Cartesian Product

Convenient, natural additions to the set of operations make

- Set Intersection
- Natural Join
- Division
- Assignment

Tuple calculus:

- The tuple relational calculus is a nonprocedural language. (The relational algebra was procedural). We must provide a formal description of the information desired.
- A query in the tuple relational calculus is expressed as

  $$\{t \mid P(t)\}$$

  i.e., the set of tuples t for which predicate P is true.
- We also use the notation
  - $t[a]$ to indicate the value of tuple t on attribute a.
  - $t \in r$ to show that tuple t is in relation r.

SQL:

- SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
- SQL statements are used to perform such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.
- Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard

SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

Integrity constraints:

- Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency.
- A form of integrity constraint with E-R models:
  - **Key declarations:** stipulation that certain attributes form a candidate key for the entity set.
  - **Form of a relationship:** mapping cardinalities 1-1, 1-many and many-many.
- An integrity constraint can be any arbitrary predicate applied to the database. They may be costly to evaluate, so we will only consider integrity constraints that can be tested with minimal overhead.

Normal Forms:

- Database normalization is the process of making the data in a database available in the most organized way possible.
- When you're normalizing a database, there are two things you need to consider: whether the information in the database has internal redundancies, and whether the dependencies across the different tables in the database are logically organized.

File organization:

- A storage hierarchy of mixed storage media
  - Primary storage: main memory, cache memory
  - Secondary storage: magnetic disks, optical disks, and tapes.
- Volatile versus non-volatile storage
- On-line versus off-line devices
- The process of **physical database design** involves choosing from among the options the particular data organization techniques that best suit the given application requirements.
- DBMS system implementers must study data organization techniques so that they can implement them efficiently and thus provide the DBA and users of the DBMS with sufficient options.

- **Primary file organizations** determine how the records of a file are physically placed on the disk, and hence how the records can be accessed.
  - A **heap file** (or **unordered file**) places the records on disk in no particular order by appending new records at the end of the file.
  - A **sorted file** (or **sequential file**) keeps the records ordered by the value of a particular field (called the sort key).
  - A **hashed file** uses a hash function applied to a particular field (called the hash key) to determine a record's placement on disk.
  - Other primary file organizations, such as B-trees, R-trees, Quad-trees, etc., use **tree structures**.

Note: The terminology may carry different meaning in different context. Sequential files in COBOL, for example, are unordered files.

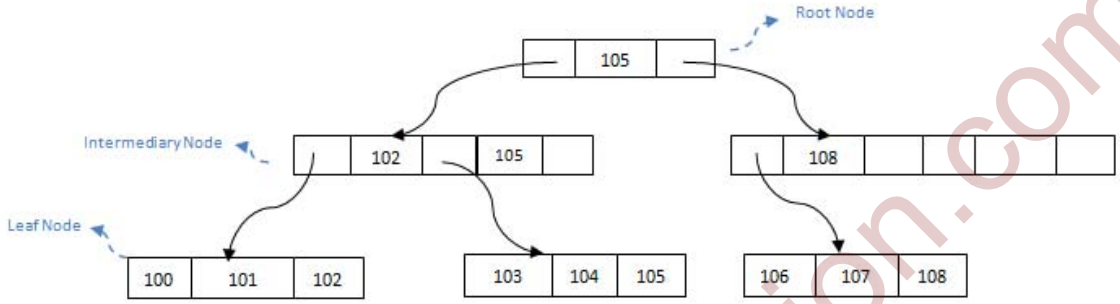
- A **secondary organization** or **auxiliary access structure** allows efficient access to the records of a file based on alternate fields than those that have been used for the primary file organization. Most of these exist as **indexes.**
- **Exercise:** Given a relation called STUDENT, what would be an appropriate primary file organization to store the records? If you choose to store the file as a sorted file, which attribute should be the sort key? Justify your answer.
- **Exercise (Cont.):** Is there a universal rule in making the above decisions?

Concepts of B+ Tree

B+ tree is a (key, value) storage method in a tree like structure. B+ tree has one root, any number of intermediary nodes (usually one) and a leaf node. Here all leaf nodes will have the actual records stored. Intermediary nodes will have only pointers to the leaf nodes; it not has any data. Any node will have only two leaves. This is the basic of any B+ tree.

Consider the STUDENT table below. This can be stored in B+ tree structure as shown below. We can observe here that it divides the records into two and splits into left node and right node. Left node will have all the values less than or equal to root node and the right node will have values greater than root node. The intermediary nodes at level 2 will have only the pointers to the leaf nodes. The values shown in the intermediary nodes are only the pointers to next level. All the leaf nodes will have the actual records in a sorted order.

STUDENT		
STUDENT_ID	STUDENT_NAME	ADDRESS
100	Joseph	Alaiedon Township
101	Allen	Fraser Township
102	Chris	Clinton Township
103	Patty	Troy
104	Jack	Fraser Township
105	Jessica	Clinton Township
106	James	Troy
107	Antony	Alaiedon Township
108	Jacob	Troy



Transactions and concurrency control:

- A transaction is a sequence of operations such that the database is in a consistent (or correct) state both before and after the transaction, but may be in an inconsistent state during the transaction.
- **Example:** Transfer $100 from account A to account B.

  Begin transaction

  Read A

  A.amount := A.amount - 100

  Write A

  Read B

  B.amount := B.amount + 100

  Write B

  End transaction
- Required properties:
  - Atomicity

  Either all the operations of a transaction are performed or none are.

- Durability

  All (committed) changes to the database persist - no matter what.
- Typical transaction structure:

  Begin transaction

Read input message

Perform processing against the database

If successfully completed

then send output message(s) and COMMIT

else ROLLBACK and send error message

EndIf

End transaction

– COMMIT

Signals successful completion of a transaction to the DBMS, which frees all locks, if any, and makes all changes permanent and visible to other.

– ROLLBACK

Signals unsuccessful completion of a transaction to the DMBS, which undoes all changes made by the transaction. If a transaction is aborted by the system, an implicit rollback must be issued on its behalf.