SCHEMA REFINEMENT and NORMAL FORMS

Session 3:**Reasoning About Functional Dependencies**

Reasoning About Functional Dependencies

- In database design, after identifying functional dependencies (FDs) during the conceptual design phase, it's important to analyze and refine these dependencies to ensure a robust and efficient schema.
- The reasoning about FDs involves understanding how a set of FDs can imply additional FDs and how to use these implications to refine the schema.

Implication of Functional Dependencies:

- Given a set of FDs over a relation schema R , additional FDs may be implied.
- For instance, consider a relation schema `Workers(ssn, name, lot, did, since)` with the following FDs:
  - ssn->did (since `ssn` is a key)
  - did->lot} (given)
- From these, we can infer that ssn->lot also holds. This is because if two tuples have the same `ssn`, they must have the same `did` (from the first FD), and if they have the same `did`, they must also have the same `lot` (from the second FD). Therefore, ssn->lot is implied.
- An FD f is said to be implied by a set F of FDs if f holds on every relation instance that satisfies all FDs in F .

Closure of a Set of FDs:

- The closure of a set of FDs F , denoted F^+, is the set of all FDs implied by F .
- To infer the closure of F , we use Armstrong's Axioms, which are three fundamental rules that allow us to generate all FDs in F^+ when applied repeatedly to F .

Closure of a Set of FDs:

- **Armstrong's Axioms:**

  - **Reflexivity:** If $X \supseteq Y$, then $X \rightarrow Y$.
  - **Augmentation:** If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z.
  - **Transitivity:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

- These axioms are sound (they generate only valid FDs) and complete (they generate all possible FDs in the closure F^+ .

- **Additional Inference Rules**
- In addition to Armstrong's Axioms, some additional rules, though not essential, are often used because they simplify the process:

  - Union: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
  - Decomposition: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.

- These additional rules can be derived from Armstrong's Axioms.

- **Example of Inference:**
- Consider a relation schema `ABC` with the FDs A->B and B->C:
  - - From transitivity, A->C.
  - From augmentation, AC->BC, AB->AC, AB->CB, and so on.
- - Using reflexivity, we generate trivial dependencies like A->A, B->B, etc.
- Now, consider a more complex example with the `Contracts` schema:
- Contracts(contractid, supplierid, projectid, deptid, partid, qty, value)` denoted as `CSJDPQV`.
- Given the following FDs:
  - 1. C->CSJDPQV (contract id is a key)
  - 2. JP->C (a project purchases a given part using a single contract)
  - SD->P (a department purchases at most one part from a supplier)
- We can infer additional FDs using Armstrong’s Axioms:
  - - From JP->C and C->CSJDPQV, by transitivity, JP->CSJDPQV .
  - - From SD-> P, by augmentation, SDJ->JP.
  - - From SDJ->JP and JP->CSJDPQV , by transitivity, SDJ->CSJDPQV .

Attribute Closure:

- To check if a specific FD :X->Y is implied by a set of FDs F , instead of computing the entire closure F^+ , we can compute the attribute closure of X with respect to F .
- The attribute closure X^+ is the set of all attributes A such that X->A can be inferred from F.

Attribute Closure:

- **Algorithm for Computing Attribute Closure:**
-

$$
\begin{aligned}
& \mathit{closure} = X; \\
& \text{repeat until there is no change: } \{ \\
& \qquad \text{if there is an FD } U \rightarrow V \text{ in } F \text{ such that } U \subseteq \mathit{closure}, \\
& \qquad\quad \text{then set } \mathit{closure} = \mathit{closure} \cup V \\
& \qquad \}
\end{aligned}
$$

- This algorithm efficiently computes the set of attributes that can be functionally determined by X under the given set of FDs F .

Attribute Closure:

- Application in Finding Candidate Keys:
- The attribute closure can also be used to find candidate keys by starting with a single attribute set X and checking if the closure of X includes all attributes of the relation schema. By varying the starting attribute and the order of considering FDs, all candidate keys can be identified.