Functional Dependencies vs. Multivalued Dependencies

Here's a brief note differentiating between Functional Dependencies (FDs) and Multivalued Dependencies (MVDs), along with examples, advantages, and disadvantages:

Definition:

- **Functional Dependency (FD):** A functional dependency, denoted as $X \rightarrow Y$, holds on a relation schema R if for any two tuples t1 and t2 in any legal instance of R, if $t1[X]=t2[X]$ (i.e., they agree on all attributes in X), then $t1[Y]=t2[Y]$ (i.e., they must agree on all attributes in Y). In simpler terms, the value of attribute set X uniquely determines the value of attribute set Y.
- **Multivalued Dependency (MVD):** A multivalued dependency, denoted as $X \twoheadrightarrow Y$, holds on a relation schema R if for any two tuples t1 and t2 in any legal instance of R such that $t1[X]=t2[X]$, then there exist tuples t3 and t4 in R such that:
  - $t3[X]=t1[X]=t2[X]$ and $t3[Y]=t1[Y]$ and $t3[R-X-Y]=t2[R-X-Y]$
  - $t4[X]=t1[X]=t2[X]$ and $t4[Y]=t2[Y]$ and $t4[R-X-Y]=t1[R-X-Y]$ Essentially, if we fix the value of X, the set of values for Y is independent of the set of values for R–X–Y.

Example:

Consider a relation CourseRegistration(StudentID, Course, Instructor).

- **Functional Dependency:** StudentID, Course ⟶ Instructor. This means for a specific student enrolled in a specific course, there is only one instructor.
- **Multivalued Dependency:** StudentID ↠ Course. This implies that for a given StudentID, the set of courses they are enrolled in is independent of the set of instructors teaching those courses. If a student is enrolled in 'Database' and 'Operating Systems', and the instructors are 'Prof. Smith' and 'Dr. Jones', then we should see combinations like (StudentID, Database, Prof. Smith), (StudentID, Database, Dr. Jones), (StudentID, Operating Systems, Prof. Smith), and (StudentID, Operating Systems, Dr. Jones) if the instructor for one course doesn't restrict the instructor for another course for the same student.

Advantages:

- **Functional Dependencies:**
  - Fundamental for defining primary keys and ensuring data integrity by enforcing constraints on attribute values.
  - Crucial for achieving 1NF, 2NF, and 3NF, reducing data redundancy and

update anomalies.
- **Multivalued Dependencies:**
  - Help in identifying and removing redundancies that cannot be addressed by FDs alone, particularly in relations with multi-valued attributes.
  - Lead to the Boyce-Codd Normal Form (BCNF) and Fourth Normal Form (4NF), resulting in more normalized and cleaner database designs.
  - Capture more complex data relationships beyond simple one-to-one or one-to-many associations.

Disadvantages:

- **Functional Dependencies:**
  - May not be sufficient to address all types of data redundancies, especially those arising from independent multi-valued attributes.
- **Multivalued Dependencies:**
  - Can be more complex to identify and understand compared to FDs.
  - Decomposition to satisfy 4NF might result in a larger number of smaller tables, potentially increasing the complexity of queries involving attributes from multiple original relations (though this is often outweighed by the benefits of reduced redundancy).

Additional Points:

- Every functional dependency is also a multivalued dependency (if $X \rightarrow Y$, then $X \twoheadrightarrow Y$). However, the converse is not always true.
- MVDs typically arise when a single key attribute determines multiple independent sets of values for other attributes.[1]
- The concept of MVDs is essential for achieving higher normal forms (beyond 3NF) and designing databases that are free from certain types of update anomalies related to multi-valued attributes.[2]
- Understanding both FDs and MVDs is crucial for effective database normalization and ensuring data integrity and efficiency. They guide the process of decomposing relations into smaller, well-structured tables.