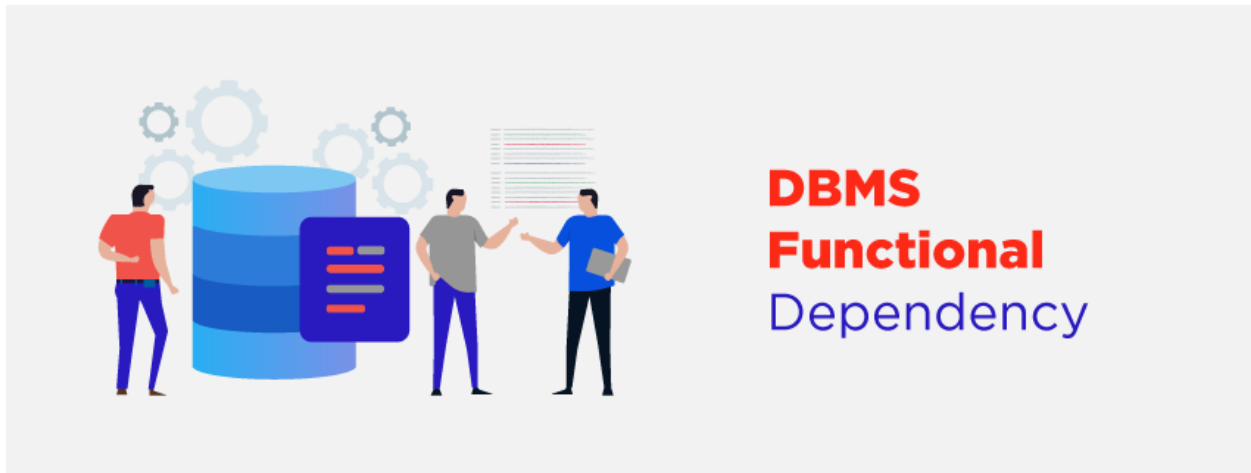


Functional Dependency in DBMS



Just like the name suggests, a Functional dependency in DBMS refers to a relationship that is present between attributes of any table that are dependent on each other. E. F. Codd introduced it, and it helps in avoiding data redundancy and getting to know more about bad designs.

What is Functional Dependency in DBMS?

Let us try to understand the concept thoroughly. If X is a relation that has attributes P and Q, then their functional dependency would be represented by \rightarrow (arrow sign)

Thus, here, the following would represent the functional dependency between the attributes using an arrow sign:

$P \rightarrow Q$

In this case, the left side of this arrow is a Determinant. The right side of this arrow is a Dependent. P will be the primary key attribute, while Q will be a dependent non-key attribute from a similar table as the primary key. It shows that the primary key attribute P is functionally dependent on the non-key attribute Q. In simpler words, If the column P attribute of a table identifies the column Q



attribute of the very same table uniquely, then the functional dependency of column Q on column P is symbolised as $P \rightarrow Q$.

Example

Let us look at an example that makes it easier to comprehend functional dependency.

Suppose we have a <Student> table with two separate attributes – Stu_Id and Stu_Name.

Stu_Id = Student ID

Stu_Name = Student Name

The Stuld is our primary key. And Stuld here identifies the StuName attribute uniquely. It is because if someone wants to know the student’s name, then you need to have the Stuld at first.

Stu_Id	Stu_Name
11	Marketing
22	HR
33	Finance
44	Accounting
55	Sales
66	Telecom

The functional dependency given above between Stu_Id and Stu_Name can be specified as Stu_Id is functionally dependent on Stu_Name.

Stu_Id -> Stu_Name



Types of Functional Dependency

The Functional Dependencies are of four major types:

1. Trivial FD
2. Non-Trivial FD
3. Completely Non-Trivial FD

1. Trivial Functional Dependency

A trivial functional dependency occurs when Q is a subset of P in –

$P \rightarrow Q$

For example,

Let us now consider the same <Section> table with two separate attributes to comprehend the trivial dependency's concept. Here is a trivial functional dependency because SecId is a subset of SecId and SecName.

$\{ \text{Sec_Id}, \text{Sec_Name} \} \rightarrow \text{Sec Id}$

2. Non –Trivial Functional Dependency

A non-trivial functional dependency occurs when Q is not a subset of P in –

$P \rightarrow Q$

For example,

$\text{Sec_Id} \rightarrow \text{Sec_Name}$

The FD given above is a functional dependency that is of a non-trivial type since SecName is not a subset of SecId.



3. Completely Non-Trivial Functional Dependency

A completely non-trivial functional dependency occurs when $P \cap Q$ is null in –

$P \rightarrow Q$

Functional Dependency's Armstrong's Axioms Property

William Armstrong developed Armstrong's Axioms property in 1974 to reason about FDs. This property suggests those rules that hold true only when the following ones are satisfied:

Augmentation

This last rule suggests that $PR \rightarrow QR$, if $P \rightarrow Q$

Reflexivity

$P \rightarrow Q$, if Q is a subset of P .

Transitivity

If $P \rightarrow Q$ and $Q \rightarrow R$, then $P \rightarrow R$ i.e. a transitive relation.

Practice Questions on Functional Dependency in DBMS

1. When an FD is reflexive, Y is a subset of X and X refers to the set of attributes, then:

- A. $Y \rightarrow X$ holds
 - B. $X \rightarrow Y$ holds
 - C. $XY \rightarrow Z$ holds
 - D. None of the above
- Answer: B. $X \rightarrow Y$ holds**



When an FD is reflexive, Y is a subset of X , and X is the set of attributes, then $X \rightarrow Y$ would always hold. It is known as the Armstrong axioms reflexivity rule.

2. If $P \rightarrow Q$, $P \rightarrow R$, then which of these would be true?

- A. $P \rightarrow QR$
- B. $P \rightarrow Q$
- C. $P \rightarrow R$
- D. All of the above

Answer: D. All of the above

3. If $A(X, Y, Z)$ refers to a relation, then which of these doesn't have a lossless join type of dependency that preserves BCNF decomposition?

- A. $P \rightarrow Q$, $Q \rightarrow RS$
- B. $P \rightarrow Q$, $Q \rightarrow R$, $R \rightarrow S$
- C. $PQ \rightarrow R$, $R \rightarrow PS$
- D. $P \rightarrow QRS$

Answer: A. $P \rightarrow Q$, $Q \rightarrow RS$

FAQs

What are functional dependencies in DBMS?

The functional dependency refers to a constraint that would specify the relationship that is between two separate sets of attributes in which one set can determine the actual value of other sets accurately. The attribute set that is present on the left side of the given arrow P is known as Determinant. On the right side, Q is known as the Dependent.

How do you identify functional dependency?

Any functional dependency, say $P \rightarrow Q$ means that no two values of Q that are different from each other are related to the same P ever. Since the key has to be unique, even when two tuples include the similar value of any attribute(s), the key values have to be different nonetheless.

