Normalization

Functional Dependencies

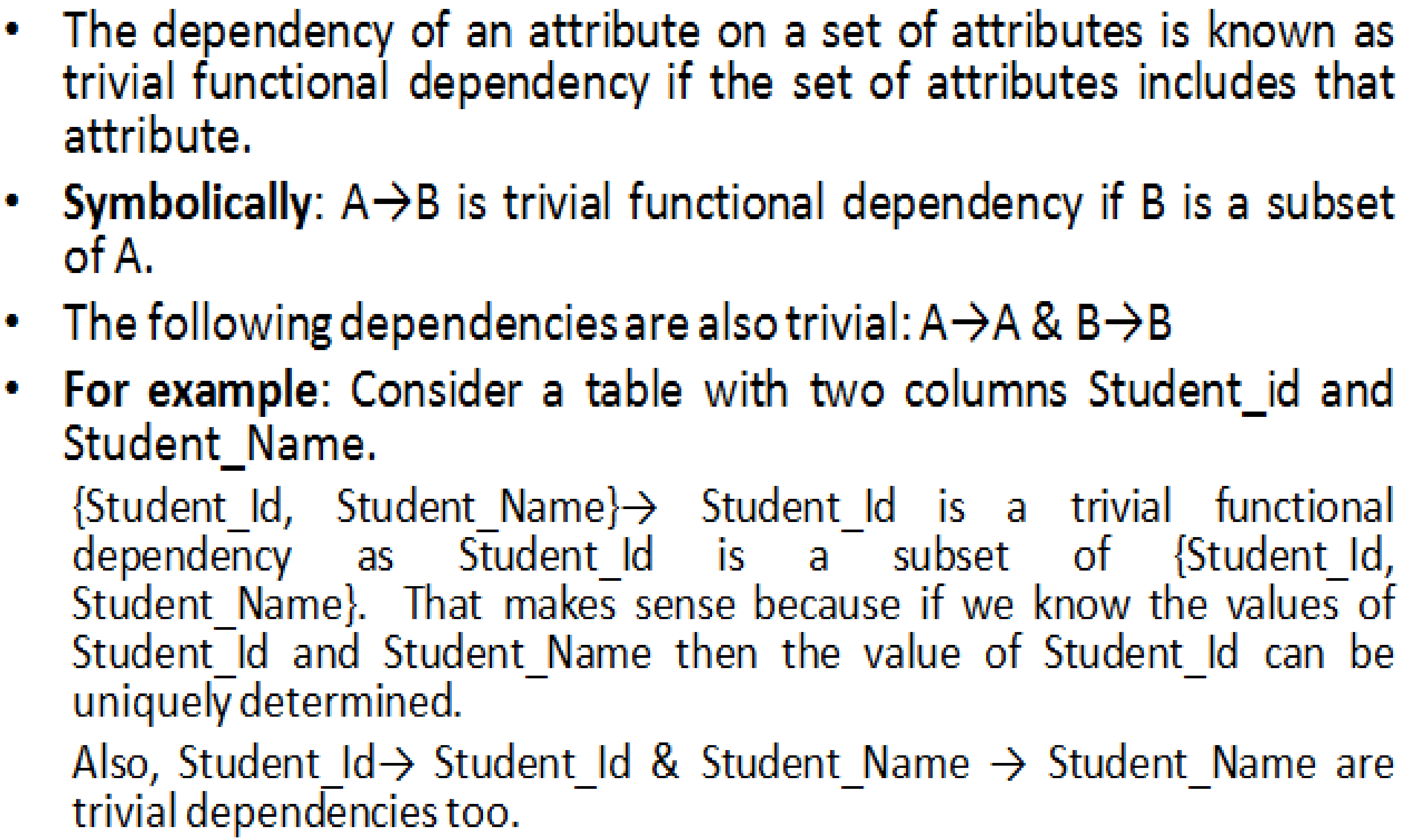
If in a table column X uniquely identifies column Y of same table then we can say that attribute Y is functionally dependent of attribute X.

It can represented as X→Y

Types of Functional Dependencies

1. Trivial functional dependency
2. non-trivial functional dependency
3. Multivalued dependency
4. Transitive dependency

Trivial Functional Dependency

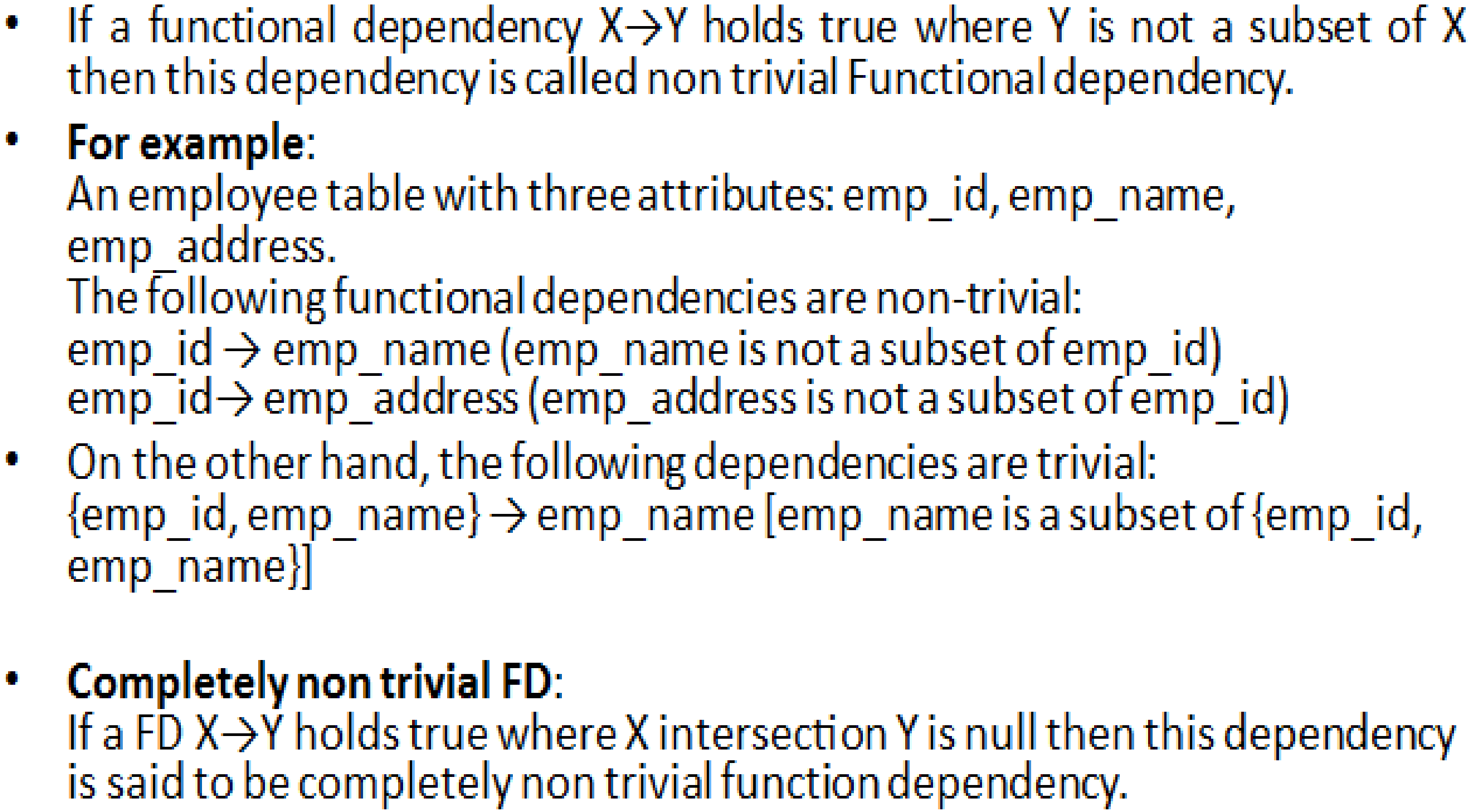


- The dependency of an attribute on a set of attributes is known as trivial functional dependency if the set of attributes includes that attribute.
- **Symbolically**: A→B is trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial: A→A & B→B
- **For example**: Consider a table with two columns Student_id and Student_Name.

{Student_Id, Student_Name}→ Student_Id is a trivial functional dependency as Student_Id is a subset of {Student_Id, Student_Name}. That makes sense because if we know the values of Student_Id and Student_Name then the value of Student_Id can be uniquely determined.

Also, Student_Id→ Student_Id & Student_Name → Student_Name are trivial dependencies too.

Non-trivial Functional Dependency

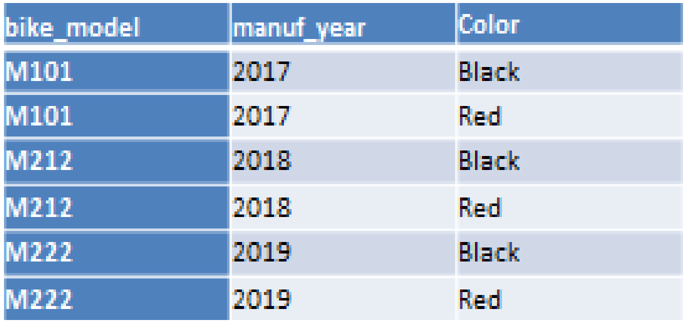


- If a functional dependency X→Y holds true where Y is not a subset of X then this dependency is called non trivial Functional dependency.
- **For example**:
  An employee table with three attributes: emp_id, emp_name, emp_address.
  The following functional dependencies are non-trivial:
  emp_id → emp_name (emp_name is not a subset of emp_id)
  emp_id→ emp_address (emp_address is not a subset of emp_id)
- On the other hand, the following dependencies are trivial:
  {emp_id, emp_name} → emp_name [emp_name is a subset of {emp_id, emp_name}]

- **Completely non trivial FD**:
  If a FD X→Y holds true where X intersection Y is null then this dependency is said to be completely non trivial function dependency.

Multivalued Dependency

- Multivalued dependency occurs when there are more than one **independent** multivalued attributes in a table.
- **For example**: Consider a bike manufacture company, which produces two colors (Black and white) in each model every year.



bike_model	manuf_year	Color
M101	2017	Black
M101	2017	Red
M212	2018	Black
M212	2018	Red
M222	2019	Black
M222	2019	Red

- Here columns manuf_year and color are independent of each other and dependent on bike_model. In this case these two columns are said to be multivalued dependent on bike_model. These dependencies can be represented like this:
- bike_model →→ manuf_year
- bike_model →→ color

Transitive Dependency

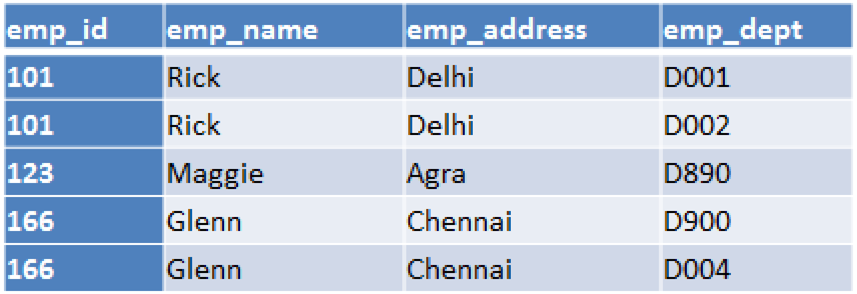
- A functional dependency is said to be transitive if it is indirectly formed by two functional dependencies.

  For e.g.

- X → Z is a transitive dependency if the following three functional dependencies hold true:
  - X→Y
  - Y does not →X
  - Y→Z

Normalization

- Organizing the data in database so that data redundancy, insertion anomaly, update anomaly & deletion anomaly can be avoided is called normalization.
- If a database is not normalized there will be three type of anomalies which occurs:
  1. Insertion anomaly
  2. Update anomaly
  3. Deletion anomaly
- Example-



emp_id	emp_name	emp_address	emp_dept
101	Rick	Delhi	D001
101	Rick	Delhi	D002
123	Maggie	Agra	D890
166	Glenn	Chennai	D900
166	Glenn	Chennai	D004

- The above table is not normalized. We will see the problems that we face when a table is not normalized.
- **Update anomaly**: In the above table we have two rows for employee Rick as he belongs to two departments of the company. If we want to update the address of Rick then we have to update the same in two rows or the data will become inconsistent. If somehow, the correct address gets updated in one department but not in other then as per the database, Rick would be having two different addresses, which is not correct and would lead to inconsistent data.
- **Insert anomaly**: Suppose a new employee joins the company, who is under training and currently not assigned to any department then we would not be able to insert the data into the table if emp_dept field doesn't allow nulls.
- **Delete anomaly**: Suppose, if at a point of time the company closes the department D890 then deleting the rows that are having emp_dept as D890 would also delete the information of employee Maggie since she is assigned only to this department.

Normal Forms

- **First normal form(1NF)-** An attribute in a table must have atomic values that means it can not have multiple values.

- **Second normal form(2NF)-** 1st NF + No non-prime attribute should have partial dependency on any candidate key of table.

- **Third normal form(3NF)-** 2nd NF + No Transitive functional dependency of non-prime attribute on any super key

Example 1-

- Let R(A,B,C,D,E) be a relation with set of functional dependencies {A →B,B →C,D →E}.
- Now we will find the candidate key that is (A,E).
- The table is in 1st NF because it has all atomic values but to be in 2nd NF all non-prime attribute should have full dependency on key not on the part of key. Here attribute B and C are only dependent on A and E is only dependent on D. so the relation is not in second NF.

Example 2-

- Let R(A,B,C,D,E) be a relation with set of functional dependencies {A →B,B →CD,D →E}.
- Now we will find the candidate key that is A.
- The table is in 1st NF because it has all atomic values but to be in 2nd NF all non-prime attribute should have full dependency on key not on the part of key. Here all attribute are fully dependent on A. so the relation is in second NF. But there is transitive dependency between A to D and A to E. So the relation is not in 3 NF.