Part 1 Normalization

Database Normalization

* **Normalization** 🡺 The Process of structuring data to minimize duplication and inconsistencies.
* The process usually involves breaking down a single Table into two or more tables and defining relationships between those tables.
* Normalization is usually done in stages, with each stage applying some rules to the types of information which can be stored in a table.

Well – Structured Relations

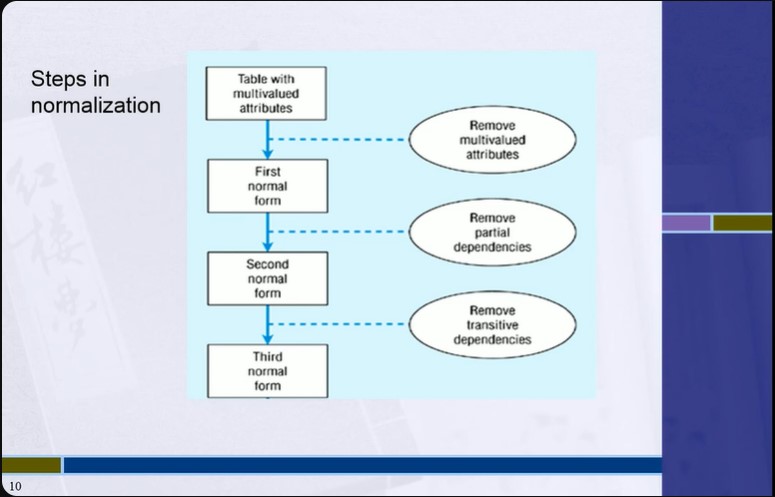
* Goal is to avoid anomalies
  + **Insertion Anomaly** – adding new row forces user to create duplicate data.
  + **Deletion Anomaly** – deleting rows may cause a loss of data that would be needed for other future rows.
  + **Modification Anomaly** – changing data in a row forces changes to other row because of duplication.
* Important Note 🡺 **A table should not have more than one entity type.**

Functional Dependency

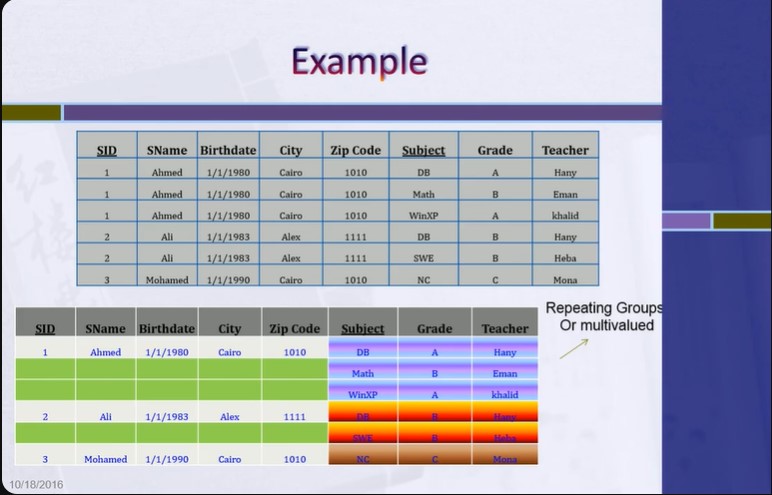
* A constraint between two attributes (columns) or two sets of columns.
* A 🡺 B if “for every valid instance of A, that value of A uniquely determines the value of B”.
* Or A 🡺 B if “existing of B depending of a value of A”.
* Some Examples:
  + Social security number determines employee name
    - (SSN 🡺 ENAME).
  + Project number determines project name and location.
    - (PNUMBER 🡺 (PNAME, PLOCATION)).
  + Employee SSN and Project Number determines the hours per week that the employee works on the project.
    - ((SSN, PNUMBER) 🡺 HOURS).

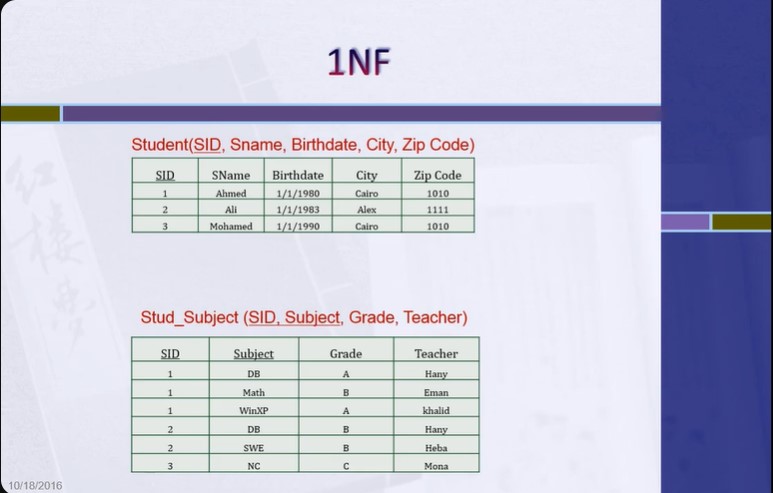
Types of Functional Dependency

* **Full Functional Dependency**
  + Attribute is fully Functional Dependency on a PK if its value is determined by the whole PK.
* **Partial Functional Dependency**
  + Attribute if has a Partially Functional Dependency on a PK if its value is determined by part of the PK (Composite Key).
* **Transitive Functional Dependency**
  + Attribute is Transitive Functional Dependency on a table if its value is determined by another non-key attribute which itself determined by PK.

Steps in Normalization

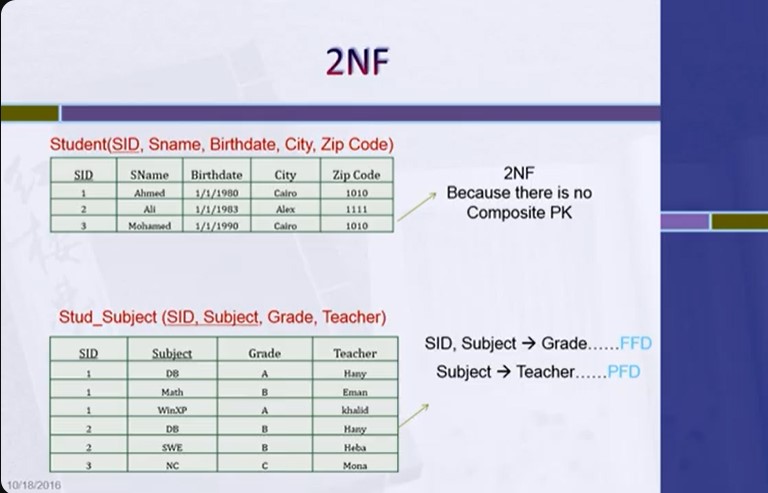
1NF (First Normal Form)

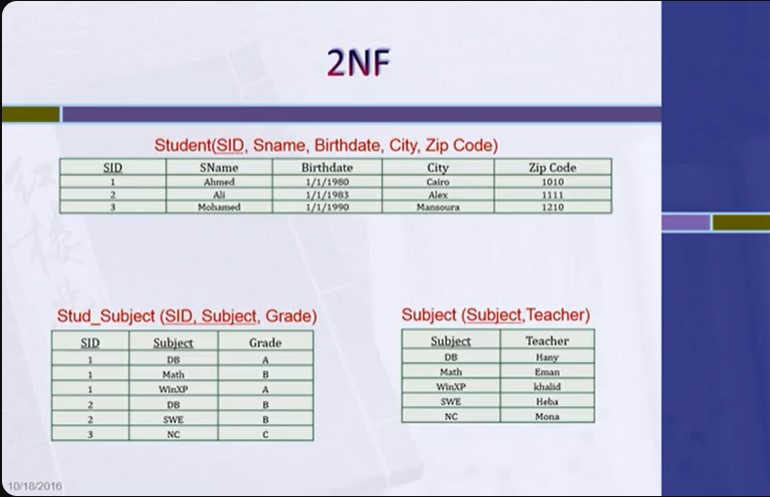
* Relation is in **first normal form** if it contains no multivalued or composite attributes.
* Remove repeating groups to a new table as already demonstrated, “carrying” the PK as a FK.
* All columns (fields) must be atomic.
  + Means 🡺 No repeating items in columns.



2NF (Second Normal Form)

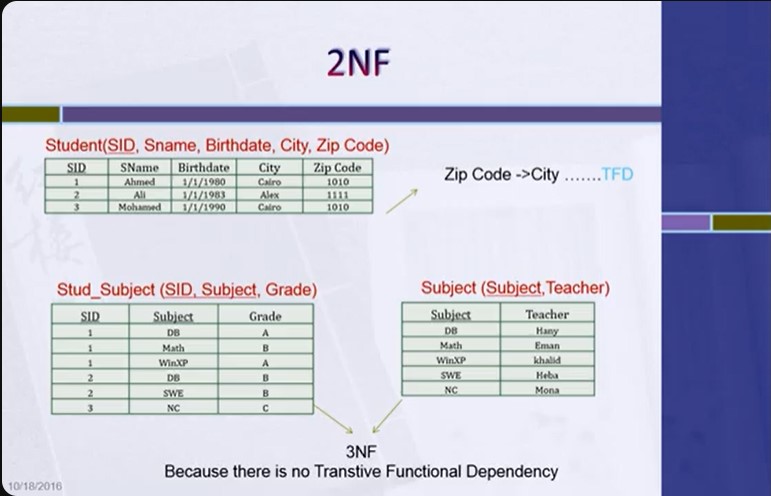
* A relation is in **second normal form** if it is in first normal form AND every non-key attribute is fully functionality dependent on the primary key.
* I.e. remove partial functional dependencies, so no non-key attribute depends on just part of the key.

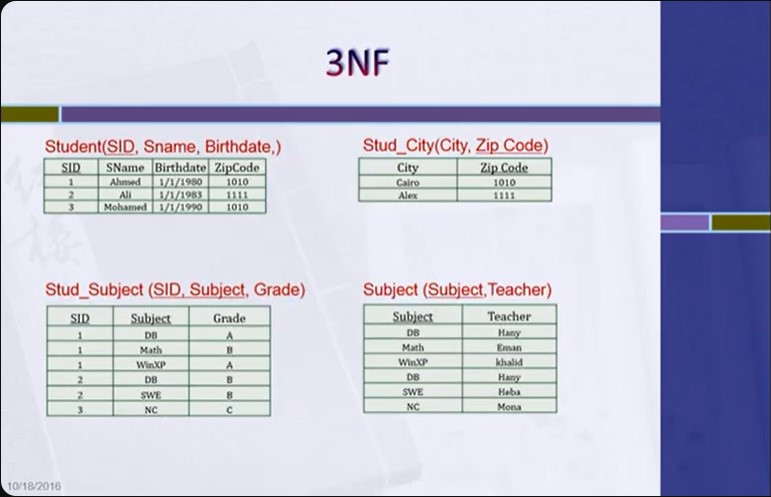




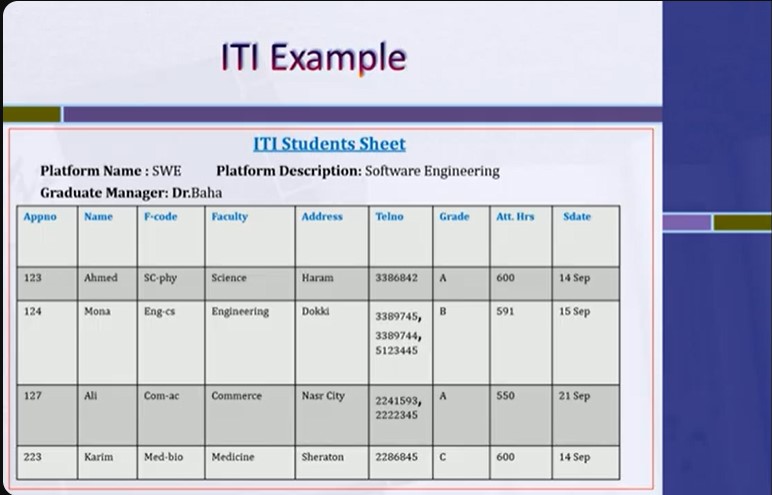
3NF (Third Normal Form)

* 2NF plus **No transitive dependences** (one attribute functionally determines a second, which functionally determines a third).

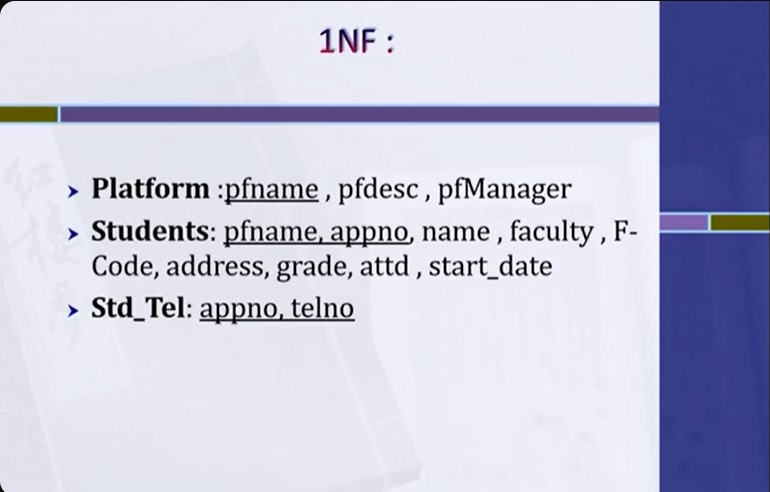


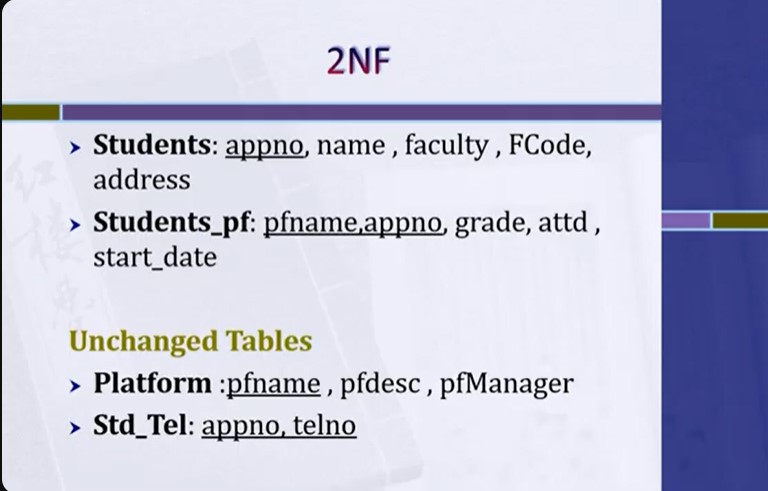


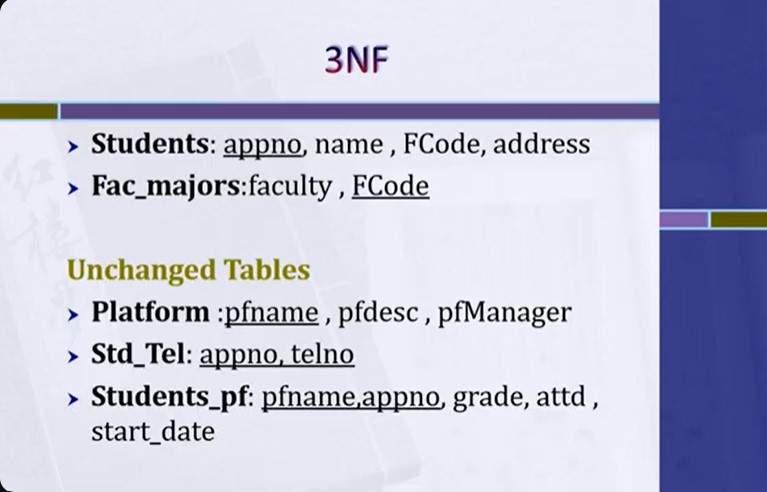
ITI Example



1NF (First Normal Form)



2NF (Second Normal Form)

3NF (Third Normal Form)