NORMALIZATION

NORMALIZATION

- Normalization is the process of decomposing a relation based on functional dependency and primary key.
- It is a step by step process to produce more efficient and accurate database design.
- Purpose is to produce an anomaly(Error) free database

NORMALIZATION

- Anomalies are Redundancy, Insertion, Deletion and updation.
- Normalized design makes the maintenance of database easier.
- Normalization applied on each table of a database design.
- Perform after the logical database design.

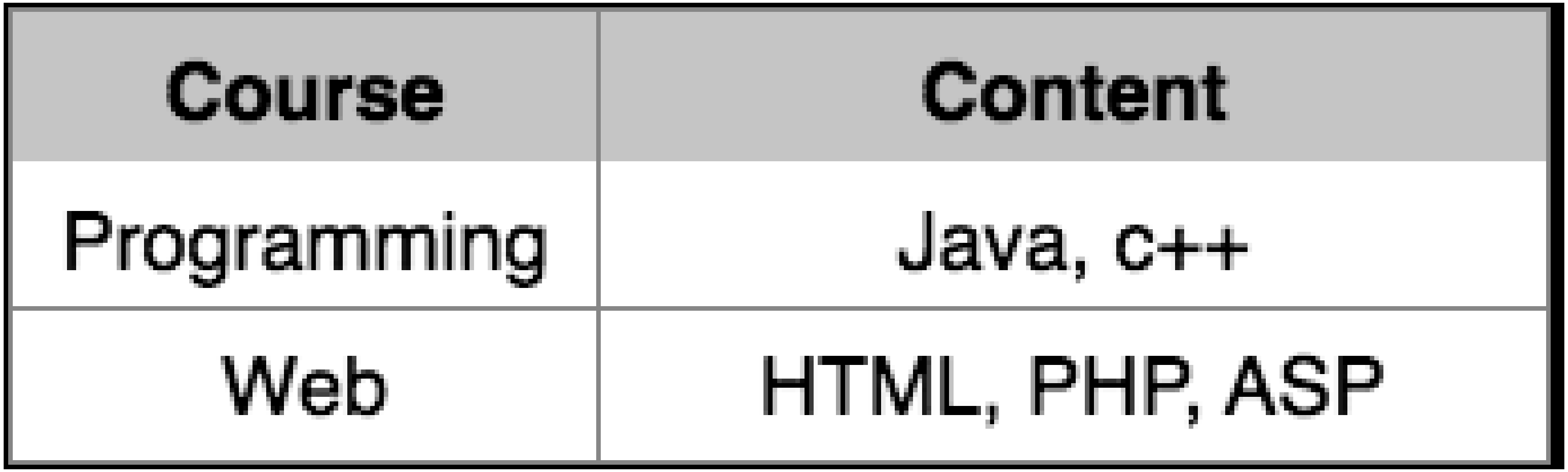
Types of Normalization

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)

First Normal Form (1NF)

- A relation is in first normal form if the values in the domain of each attribute of the relation are atomic.
- Each cell of the table must have single value
- No two rows in a table may be identical .
- Main purpose of 1NF is identify and remove the repeating groups within the table

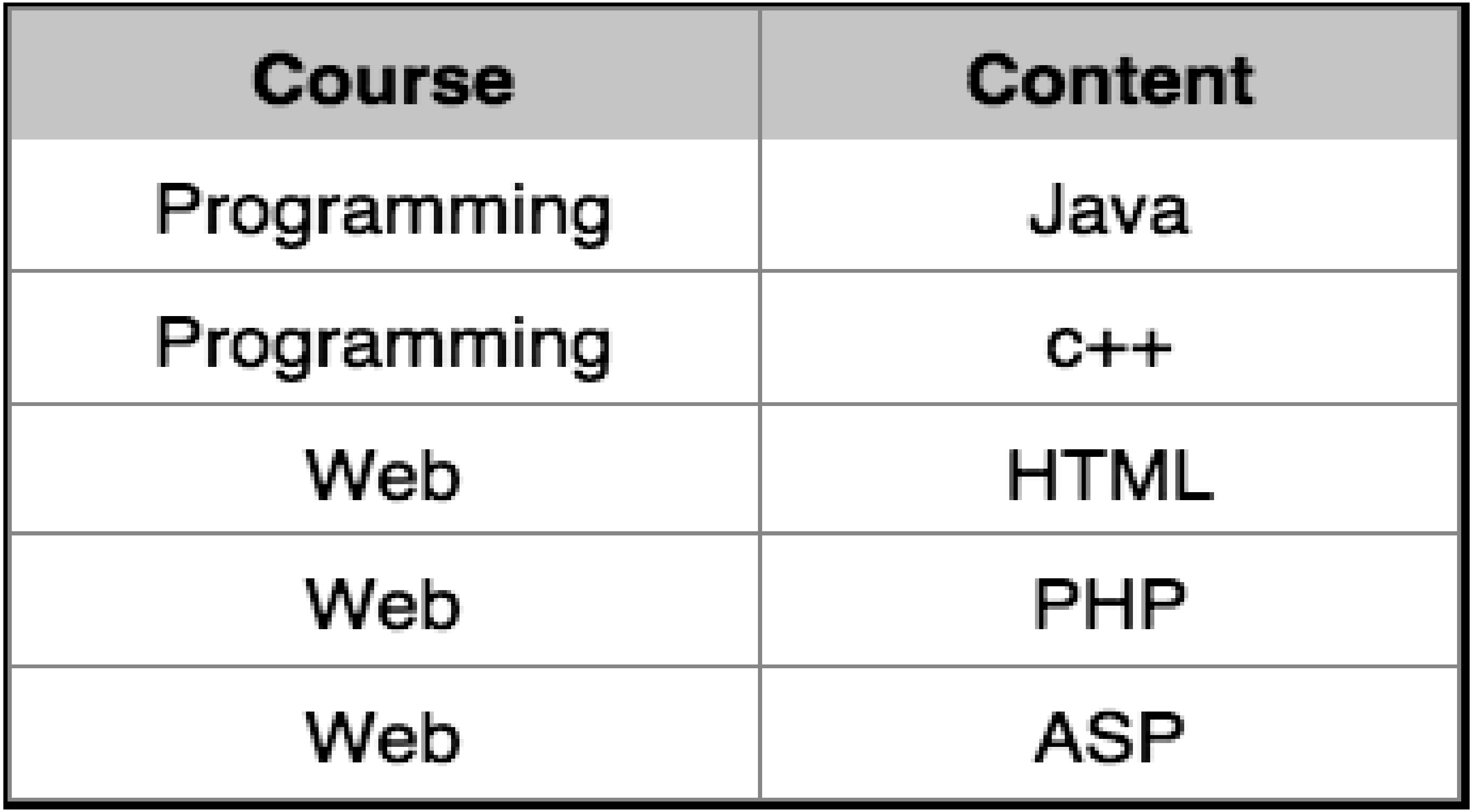
Example of 1NF



Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

Unorganized relation

Example of 1NF



Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

Relation in 1NF

Second Normal Form(2NF)

- A relation is in 2NF if it is in the first normal form and all non key attributes are fully functionally dependent on key, that is, there is no partial dependency.
- Key Attributes (Keys :- ex primary key)
- Non Key Attributes (Attributes Except key)
- Fully Functionally dependency
- Partial dependency

Fully Functional dependency

- If all the non-key attributes fully dependent on key and there is no any subset of primary key determine any non-key attribute, then it is called fully functional dependency.
- Ex:-

  Std(id, name, roll, address,)

  id -> name, roll, address,

Partial dependency

- If any proper subsets of the key determine any of the non-key attributes, then it is called partial dependency.
- Ex:- Class(crId, stId, stName, fId, room, grade)

    crId, stId -> stName, fId, room, grade

    stId -> stName

    crId -> fId, room

Example of 2NF

- Ex:- Class(crId, stId, stName, fId, room, grade)

  crId, stId -> stName, fId, room, grade

  stId -> stName

  crId -> fId, room

- Here relation Class is not in 2NF

Example of 2NF

- Relation is decomposed based on FDs

- STD(stId, stName)
- COURSE(crId, fId, room)
- CLASS(crId, stId, grade)

Each of these tables is in 2NF

Third Normal Form(3NF)

- A relation is in third normal form if it is in 2NF and there is no transitive dependency, that is, no non-key attribute is dependent on another non-key attribute.

- A relation R is in 3NF if it is in 2NF and has no transitive dependency.

Example of 3NF

- STD(stId, stName, stAdd, prName, prCrdts)
- stId -> stName, stAdr, prName
- prName -> prCrdts

STD Relation is not in 3NF

Example of 3NF

- Converting STD Relation in 3NF By decomposing STD Relation

- STD(stId, stName, stAdr, prName)
- PROGRAM(prName, prCrdts)

Each of the tables is in 3NF

Boyce-Codd Normal Form

- A relation R is in BCNF if every determinant is a candidate key.
- Every relation in BCNF is in 3NF vice - versa is not always true
- 3NF is checked in steps
- BCNF perform directly
- It is general form of 3NF

Example of BCNF

- FACULTY(fId, dept, office, rank, dateHired)
- fId ,dept -> office, rank, dateHired
- Offce -> dept

- Relation is in 3NF but not in BCNF because office is not a candidate key.

Example of BCNF

- We decompose the table again to bring it into BCNF

- FACULTY(fId,office, rank, dateHired)
- OFFICE(office, dept)

These tables are in 3NF and BCNF