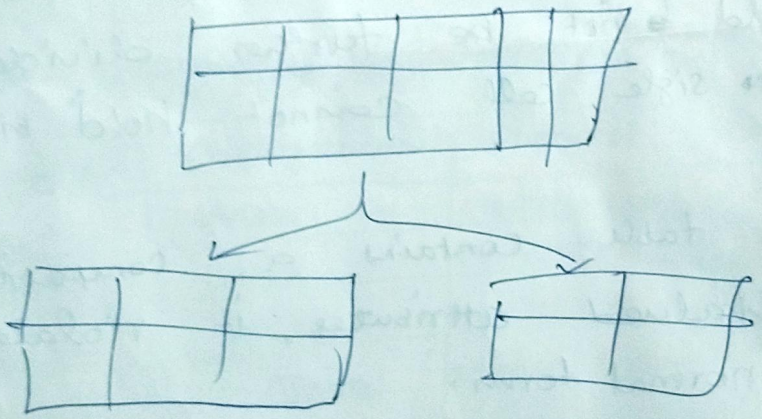
Normalization

* It is a technique of organizing data in the database. It is a systematic approach to decompose tables to eliminate data redundancy (repetition), and ~~remove~~ and set up the relationships between tables.

* Improves Data Integrity
* It is a multistep process that puts data into tabular form removing the duplicated data from its relational tables.



every table in the database has to be in the normal form

Advantages:

* used to eliminate inconsistent and repeated data
* Used to eliminate data anomalies.
  * insertion anamalies

* updation anomaly
* deletion anomaly.

Types of normal Form

1. 1 FN
2. 2 NF
3. 3 NF
4. BCNF
5. 4NF
6. 5 NF

First Normal Form (1NF):

* problem of atomicity
  ↓
  means values in the table should ~~b~~ not be further divided
  (ie) a single cell cannot hold multiple values.

If a table contains a composite or multivalued attributes, it violates the first normal form.

Functions performed in First Normal form:

1. Removes repeating groups from the table.
2. Create a separate table for each set of related data
3. Identify each set of related data with a primary key.

Ex:

Consider the Employee table

EmployeeID	Employee Name	phone No	Salary
EID 001	Alex	+919750038854 +919842626725	55,000
EID 002	David	+91 7548 8875	50000
EID 003	John	+91 6842578724	22900
EID 004	Barry	+91672548447 +917250787	30000

phone_no column contains two values. so it violates the ~~1NF~~ 1NF condition.

So Apply the 1NF, in the above table, we get the following form.

EmployeeID	Employeename	phoneno	Salary
EID001	Alex	+919750038854	55,000
EID001	Alex	+919842626725	55,000
EID002	David	+91 7548 8875	50,000
EID003	John	+91 6842 578724	22900
EID004	Barry	+91672 54844	30,000
EID004	Barry	+917250787 87	30,000

In this table each ~~cell~~ tuple has distinct values, ie no cell have multiple values. Thus the 1NF is achieved.

Second Normal Form (2 NF):

When a table is said to be in 2NF only when it fulfills the following conditions.

* It has to be in 1st Normal form
* The table also should not contain any partial dependency.

partial dependency:

The proper subset of a Candidate key determines a non-prime attribute.

Non-prime attribute:

Attributes that form a Candidate key in a table are called prime attributes. The rest of the attributes of the relation are called non-prime attributes.

Ex: prime attributes: EmployeeID, Dept-ID
Non-prime a : office location

Ex: Consider this table

EmployeeID	Dept-ID	Office location
EID 001	EDID 1	pune
EID 002	EDID 2	Chennai
EID 003	EDID 3	mumbai
EID 004	EDID 4	Delhi

This table has a composite primary key consists of EmployeeID and Dept-ID and the non-primary key is OfficeLocation.

Here OfficeLocation only depends on Department location (Dept-ID) which is only the part of the primary key. Therefore this table doesnot satisfy the second Normal Form (2NF).

To bring this table in 2NF, split the table into two tables such that the first table has EmployeeID and Dept-ID and the second table has Dept-ID and OfficeLocation columns.

EmployeeID	Dept-ID

Dept-ID	OfficeLocation

From these tables, we have removed the partial functional dependency because the OfficeLocation is fully dependent on the primary key attribute Dept-ID. ~~still similarly~~

Ex:/

Third Normal Form (3NF):

The 3NF is a normal form that is used in normalizing the table to reduce the duplication of data and ensure referential integrity.

The following condition has to be satisfied in 3NF:

* The table has to be in 2NF
* There should be no transitive dependency for non-prime attributes
  ie no non-prime attribute is transitively dependent on any non-prime attribute which depends on other non-prime attributes

ex: if C is dependent on B and in turn B is dependent on A, then transitively C is dependent on A.

if $C \rightarrow B$
$B \rightarrow A$
then $C \rightarrow A$

This should not happen on 3NF.

All the non-prime attributes must depend only on the prime attributes.

These are the two necessary conditions that needs to be attained in 3NF.

Ex:

StudentID	StudentName	SubjectID	subject	Address
SID 001	Rama	50cs001	SQL	Chennai
SID 002	Ramu	50cs002	DBMS	Ooty
SID 003	David	50cs003	C++	Bangalore
SID 004	Michale	50cs004	Java	mumbai

In this table StudentID determines SubjectID and SubjectID determines subject. ≐

∴ StudentID determines subject via SubjectID.

This implies that we have transitive functional dependency and this table doesnot satisfy 3NF.

In order to achieve the 3NF, we have to divide the table into 2 tables. The first table contains StudentID, StudentName, SubjectID and address. Here primary key is studentID. All the non-prime attributes depends only on the studentID primary key.

StudentID	StudentName	SubjectID	Address

The second table contains SubjectID and subject. Here, The subject is dependent only on the SubjectID primary key.

SubjectID	Subject

∴ Now all the non-key attributes are fully functionally dependent only on the primary key

Boyce codd Normal form (BCNF):

It is also known as 3.5 NF. It is a higher version of 3NF and was developed by Raymond F boys and Edgar F. codd to address certain types of ~~and~~ anomalies which were not dealt with 3NF.

The table in BCNF should ~~sto~~ satisfy the following conditions:

* It has to be in 3rd Normal Form (3NF)

* Every functional dependency $A \rightarrow B$, then A has to be the super key of that particular table

Ex:

Student_ID	Subject	Professor
SID 001	SQL	Prof. Anand
SID 002	DBMS	Prof. Karthi
SID 003	C++	Prof. Lokesh
SID 004	Java	Prof. Sheela

One student can enroll for multiple subjects. and there can be multiple professors can teach one subject.

For each subject, a professor is assigned to the students.

These are the necessary conditions on the table.

In this table, all the Normal forms are satisfied except BCNF.

Here, StudentID and subject forms the primary key.
Here subject is the prime attribute.

Consider the dependency, the professor is depending on subject,

~~Subject → Professor~~ professor → subject

Here subject is prime attribute and professor is non-prime attribute. It is not allowed in BCNF.

In order to satisfy the BCNF, divide the table into two tables.

StudentID	ProfessorID

↓
new column is created.

ProfessorID	Subject	professor

ProfessorID → Subject
ProfessorID → professor

In the second table, subject and professor depends only on professorID, which is a primary key of this table.

Thus, we have removed the non prime attribute functional dependency.

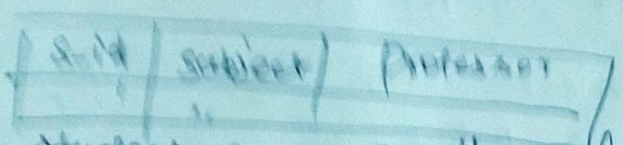
BCNF Normal Form

Functional dependency.

$A \rightarrow B$

↓

super key.



One student can enroll multiple subjects.

multiple professors can teach one subject.

For Each subject, a professor assigned to the students.

necessary conditions

S1 — Java. AB

S1 — DBMS BA.

S2 — Java AB

— SQL AC.

S-id, Subject — forms the primary key.

subject prime attribute.

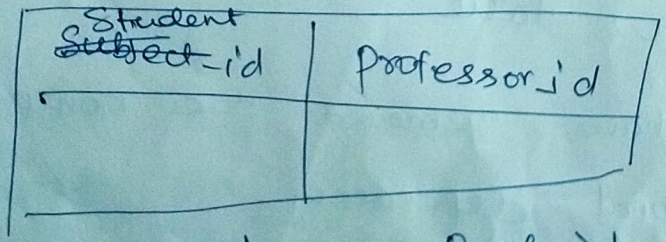
professor is depending on subject.

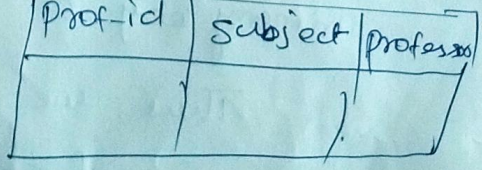
professor → subject.

but professor is non prime attribute.

This is not allowed in BCNF

divide the table into two parts.





new column Prof-id is introduced.

remove non prime attribute functional dependency.