The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Database Management Systems

ITC 304

Subject Incharge

Priyanka Patil

Associate Professor

Room No. 316

email: priyankapatil@sfit.ac.in

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Module V

Lecture 1

Relational Database Design

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Content

- Design guidelines for relational Schema
- Database tables and normalization
- The need for normalization
- Functional Dependencies
- Types of Functional Dependencies fully FD,Partial FD,trivial,non trivial, sem trivial, Armstrong axioms FD,Closure set/attribute closure

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Pitfalls in Relational-Database Design

- Relational database design requires that we find a “good” collection of relation schemas.
- A bad design may lead to
  - Repetition of Information.
  - Inability to represent certain information.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Goals Design guidelines for relational Schema

1. Avoid **redundant** data in row and columns .
2. Ensure that **relationships among** attributes are represented .
3. Facilitate the **checking of updates** for violation of database integrity constraints
4. Avoid **Update/Modification** anomalies, **Insertion** anomalies ,**Deletion** anomalies
5. Problems with **Null Values and Dangling Tuples.(**Problems when we apply joins**)**

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Example

- Example
- Consider the relation schema:

Lending-schema = (branch-name, branch-city, assets, customer-name,loan-number, amount)

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Problems

Redundancy:

. Data for branch-name, branch-city, assets are repeated for each loan that a branch makes

. Wastes space

. Complicates updating

Null values:

- Cannot store information about a branch if no loans exist .
- Can use null values, but they are difficult to handle.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

- In the given example the database design is faulty which makes the above pitfalls in database. So we observe that in relational database design if the design is not good then there will be faults in databases.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Functional Dependency

- The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

$$X \rightarrow Y$$

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Example of Functional Dependency

Assume we have an **employee table** with attributes: Emp_Id, Emp_Name, Emp_Address.

Here **Emp_Id attribute** can **uniquely** identify the Emp_Name attribute of employee table because **if we know the Emp_Id,** we can tell that employee name associated with it.

Functional dependency can be written as:

Emp_Id → Emp_Name

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Rules of Functional Dependencies/Inference Rules/Armstrong Axiom's

- Armstrong's Axioms is a set of rules.
- It provides a simple technique for reasoning about functional dependencies.
- It was developed by William W. Armstrong in 1974.
- It is used to infer all the functional dependencies on a relational database.

Various Axioms / Rules

A. Primary Rules

B. Secondary Rules

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Rules of Functional Dependencies/Inference Rules/Armstrong Axiom's

A. Primary Rules of Armstrong Axioms

1. **Reflexive rule:** If X is a set of attributes and Y is_subset_of X , then X holds a value of Y.
X -> Y

2. **Augmentation rule:** When X -> Y holds, and Z is attribute set, then XZ -> YZ also holds. That is adding attributes which do not change the basic dependencies.

3 **Transitivity rule:** This rule is very much similar to the transitive rule in algebra if X -> Y holds and Y -> Z holds, then X -> Z also holds. X -> Z is called as X functionally that determines Z.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Rules of Functional Dependencies/Inference Rules/Armstrong Axiom's(Cont..)

B. Secondary Rules

1.Union

If X holds Y and X holds Z, then X holds YZ.

If $\{X \rightarrow Y\}$ and $\{X \rightarrow Z\}$, then $\{X \rightarrow YZ\}$

2.Decomposition

If X holds YZ and X holds Y, then X holds Z.

If $\{X \rightarrow YZ\}$ and $\{X \rightarrow Y\}$, then $\{X \rightarrow Z\}$

3.Pseudo Transitivity

If X holds Y and YZ holds W, then XZ holds W.

If $\{X \rightarrow Y\}$ and $\{YZ \rightarrow W\}$, then $\{XZ \rightarrow W\}$

4.Composition

If $\{X \rightarrow Y\}$ and $\{Z \rightarrow W\}$,then XZ -> YW

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies

1. **Trivial Functional Dependency**
2. **Non-Trivial Functional Dependency**
3. **Multivalued Dependency**
4. **Transitive Dependency**

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies

1.Trivial Functional Dependency

- In Trivial Functional Dependency, a dependent is always a subset of the determinant. i.e. If X → Y and Y is the subset of X, then it is called trivial functional dependency
- X intersection Y cannot not be null.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

Example:

Consider a table with two columns Employee_Id and Employee_Name.

{Employee_id, Employee_Name} → Employee_Id is a trivial functional dependency as Employee_Id is a subset of {Employee_Id, Employee_Name}.

Also, Employee_Id → Employee_Id and Employee_Name → Employee_Name are trivial dependencies too.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

2. Non-trivial functional dependency

- Functional dependency which also known as a nontrivial dependency occurs when X->Y holds true where Y is not a subset of X.
- In a relationship, if attribute Y is not a subset of attribute X, then it is considered as a non-trivial dependency.
- When X intersection Y is NULL, then X → Y is called as complete non-trivial.

Example:

ID → Name

Name → DOB

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

Company	CEO	Age
Microsoft	Satya Nadella	51
Google	Sundar Pichai	46
Apple	Tim Cook	57

Example:

(Company} -> {CEO}

(if we know the Company, we knows the CEO name)

But CEO is not a subset of Company, and hence it's non-trivial functional dependency.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

3.Multivalued Dependency in DBMS

- Multivalued dependency occurs when two attributes in a table are independent of each other but, both depend on a third attribute.
- A multivalued dependency consists of at least two attributes that are dependent on a third attribute that's why it always requires at least three attributes.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

Example:

Car_model	Manufacturing year	Color
H001	2017	Metallic
H001	2017	Green
H005	2018	Metallic
H005	2018	Blue
H010	2015	Metallic
H033	2012	Gray

In this example, year and color are independent of each other but dependent on car_model. In this example, these two columns are said to be multivalue dependent on car_model. This dependence can be represented like this:

car_model -> Manufacturing year

car_model-> colour

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

4.Transitive Dependency in DBMS

- A Transitive Dependency is a type of functional dependency which happens when "t" is indirectly formed by two functional dependencies.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Functional Dependencies(Cont..)

Example:

Company	CEO	Age
Microsoft	Satya Nadella	51
Google	Sundar Pichai	46
Alibaba	Jack Ma	54

- {Company} -> {CEO}

(if we know the company, we know its CEO's name)

- {CEO } -> {Age}

If we know the CEO, we know the Age

- Therefore according to the rule of rule of transitive dependency:
- { Company} -> {Age} should hold, that makes sense because if we know the company name, we can know his age.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

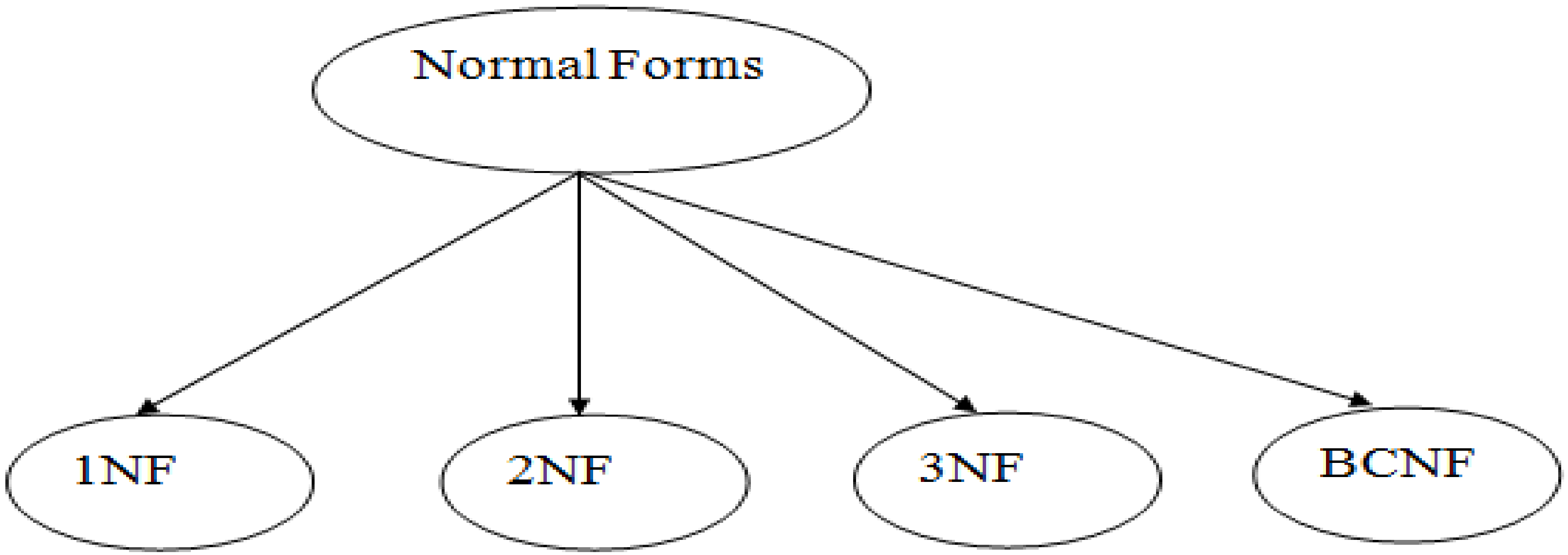
Normalization

- Normalization is the process of **organizing the data** in the database.
- Normalization is used to **minimize the redundancy** from a relation or set of relations. It is also used to **eliminate the undesirable characteristics** like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The different normal forms are used **to reduce redundancy** from the database table.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Normal Forms

- There are the four types of normal forms:



The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Need of Normalization

1. To remove the duplicate data and database anomalies from the relational table.
2. It helps to reduce redundancy and complexity by examine new data types used in the table.
3. It is useful to divide large database table into smaller tables and link them using relationship.
4. It avoids duplicate data or repeating groups into a table.
5. It reduces the chances for anomalies to occur.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Problems Without Normalization

- If a table is not properly normalized and have data redundancy then it will not only eat up **extra memory space** but will also make it **difficult to handle** and **update** the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if database is not normalized.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Example

- **Student**

Rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields **branch, hod**(Head of Department) and **office_tel** is repeated for the students who are in the same branch in the college, this is **Data Redundancy**.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization

Anomalies are inconvenient or **error-prone situations** arising when we process the tables.

There are **three different** types of anomalies:

1. Insertion Anomaly
2. Updation Anomaly
3. Deletion Anomaly

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization

1.Insertion Anomaly

An insertion anomaly is the **inability to add data** to the database **due to the absence of other data.**

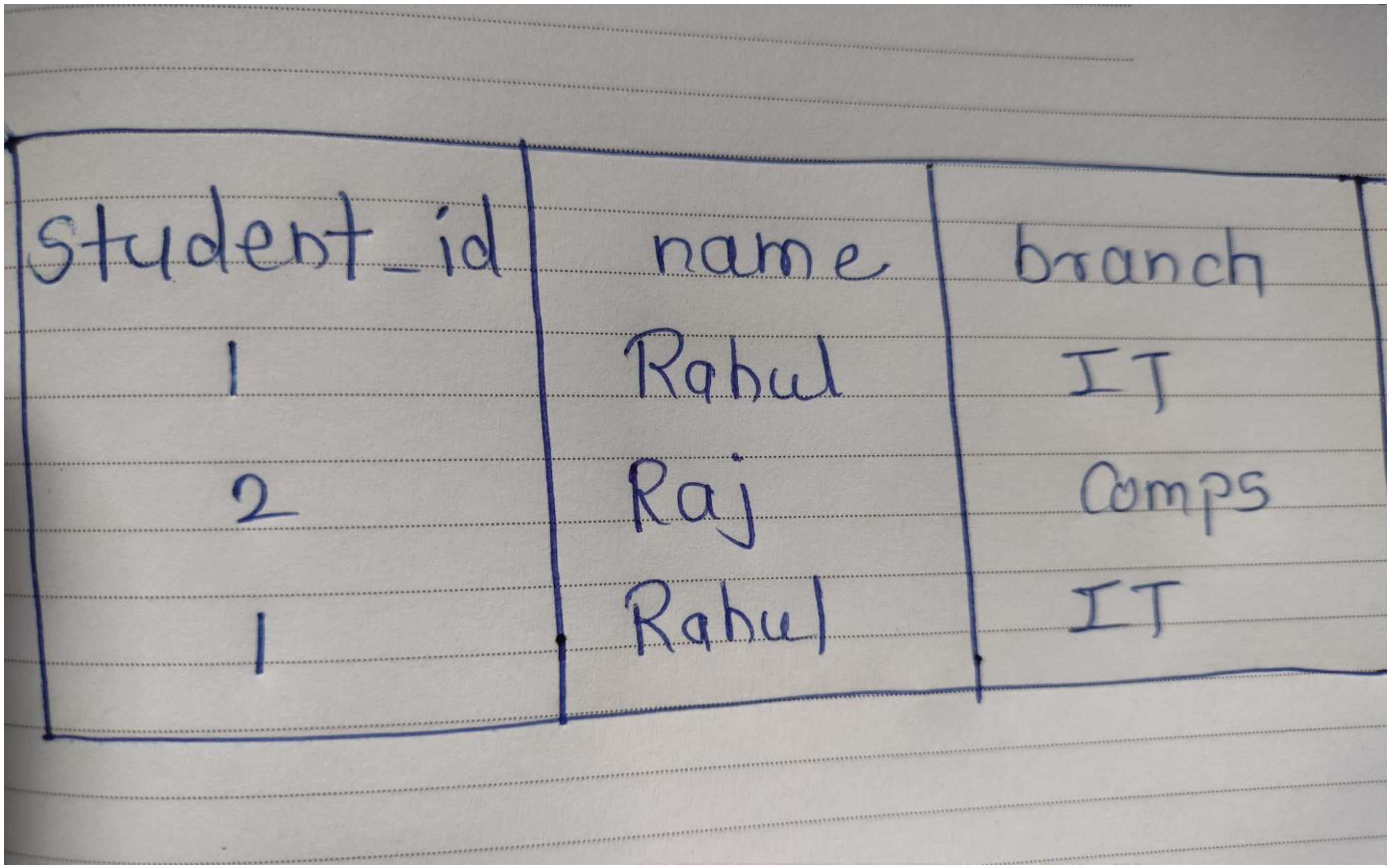
Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as **NULL**.

- Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.
- These scenarios are nothing but **Insertion anomalies**.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization

Student table with duplicacy

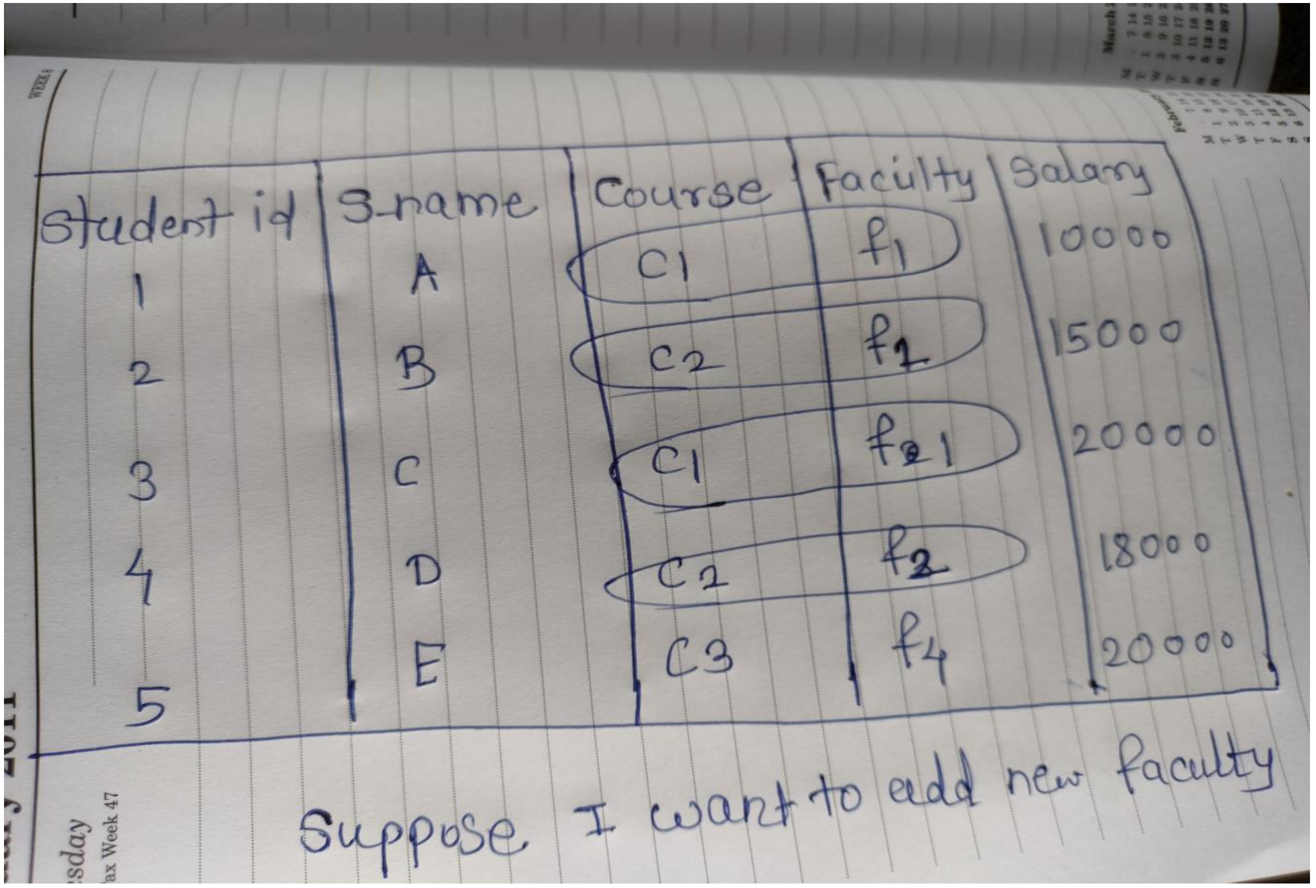


student_id	name	branch
1	Rahul	IT
2	Raj	Comps
1	Rahul	IT

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization

Student table with duplicacy



Student id	S-name	Course	Faculty	Salary
1	A	C1	f1	10000
2	B	C2	f2	15000
3	C	C1	f21	20000
4	D	C2	f2	18000
5	E	C3	f4	20000

Suppose I want to add new faculty

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization(Cont..)

2. Updation Anomaly

What if Mr. X leaves the college? or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency. This is Updation anomaly.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Anomalies without Normalization(Cont..)

3. Deletion Anomaly

In our **Student** table, two different information's are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is Deletion anomaly.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Types of Normalization

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. NO partial dependencies
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4NF	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

BCNF : A relation will be in BCNF if it is in 3NF and has no FD with non-super key determinant

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

First Normal Form (1NF)

- A relation will be in 1NF if it contains an **atomic value**.
- It states that an attribute of a table **cannot hold multiple values.** It must hold **only single-valued attribute.**
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

- The second step in Normalization is 2NF.
- A table is in 2NF, only if a relation **is in 1NF** and meet all the rules, and **every non-key attribute is fully dependent on primary key.**
- The Second Normal Form **eliminates partial dependencies** on primary keys.
- **Example**

- **<StudentProject>**

StudentID	ProjectID	StudentName	ProjectName
S89	P09	Olivia	Geo Location
S76	P07	Olivia	Geo Location
S56	P03	Ava	IoT Devices
S92	P05	Alexandra	Cloud Deployment

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

Example(Cont..)

- In the above table, we have partial dependency;
- The prime key attributes are StudentID and ProjectID.
- The non-prime attributes i.e. StudentName and ProjectName should be functionally dependent on part of a candidate key, to be Partial Dependent.
- The StudentName can be determined by StudentID, which makes the relation Partial Dependent.
- The ProjectName can be determined by ProjectID, which makes the relation Partial Dependent.
- Therefore, the <StudentProject> relation violates the 2NF in Normalization and is considered a bad database design.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

- **Example (Table converted to 2NF)**
- To remove Partial Dependency and violation on 2NF, decompose the above tables −

<StudentInfo>

StudentID	ProjectID	StudentName
S89	P09	Olivia
S76	P07	Olivia
S56	P03	Ava
S92	P05	Alexandra

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

- **Example (Table converted to 2NF)**
- To remove Partial Dependency and violation on 2NF, decompose the above tables −

<StudentInfo>

StudentID	ProjectID	StudentName
S89	P09	Olivia
S76	P07	Olivia
S56	P03	Ava
S92	P05	Alexandra

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

<ProjectInfo>

ProjectID	ProjectName
P09	Geo Location
P07	Geo Location
P03	IoT Devices
P05	Cloud Deployment

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

- **Example 2:** Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.
- **TEACHER table**

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Second Normal Form (1NF)

- In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.
- To convert the given table into 2NF, we decompose it into two tables:
- **TEACHER_DETAIL table:**

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

- **TEACHER_SUBJECT table:**

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- A relation will be in 3NF if it is in 2NF and not contain any **transitive partial dependency**.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.
- A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency X → Y.
- X is a super key.
- Y is a prime attribute, i.e., each element of Y is part of some candidate key.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- **Example:**
- **EMPLOYEE_DETAIL table:**

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- **Super key in the table above:**

1. {EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on

- Candidate key: {EMP_ID}

- Non-prime attributes: In the given table, all attributes except EMP_ID are non-prime.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The **non-prime attributes** (**EMP_STATE, EMP_CITY**) transitively dependent on super key(EMP_ID). It violates the rule of third normal form.
- That's why we need to move the EMP_CITY and EMP_STATE to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE_DETAIL table:

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively dependent on super key(EMP_ID). It violates the rule of third normal form.
- That's why we need to move the EMP_CITY and EMP_STATE to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

Example (Table violates 3NF)

<MovieListing>

Movie_ID	**Listing_ID**	**Listing_Type**	**DVD_Price** ($)
0089	007	Comedy	100
0090	003	Action	150
0091	007	Comedy	100

The above table is not in 3NF because it has a transitive functional dependency

Movie_ID -> Listing_ID
Listing_ID -> Listing_Type

Therefore, **Movie_ID -> Listing_Type** i.e. transitive functional dependency

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- **<Listing>**

Listing_ID	Listing_Type
007	Comedy
003	Action
007	Comedy

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

- **Example (Table converted to 3NF)**
- To form it in 3NF, you need to split the tables and remove the transitive functional dependency.
- **<Movie>**

Movie_ID	**Listing_ID**	**DVD_Price** ($)
0089	007	100
0090	003	150
0091	007	100

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Third Normal Form (3NF)

Unnormalized

<MovieListing>

Movie_ID	Listing_ID	Listing_Type	DVD_Price ($)
0089	007	Comedy	100
0090	003	Action	150
0091	007	Comedy	100

Normalized

<Movie>

Movie_ID	Listing_ID	DVD_Price ($)
0089	007	100
0090	003	150
0091	007	100

<Listing>

Listing_ID	Listing_Type
007	Comedy
003	Action

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

- BCNF is the advance version of 3NF. It is **stricter** than 3NF.
- A table is in BCNF if every functional dependency X → Y, X is the **super key** of the table.
- For BCNF, the table should be in 3NF, and for every FD, LHS is super key.
- **Example:** Let's assume there is a company where employees work in more than one department.
- **EMPLOYEE table**:

EMP_ID	EMP_COUNTRY	EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
264	India	Designing	D394	283
264	India	Testing	D394	300
364	UK	Stores	D283	232
364	UK	Developing	D283	549

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

- **In the above table Functional dependencies are as follows:**
- EMP_ID → EMP_COUNTRY
- EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}
- **Candidate key: {EMP-ID, EMP-DEPT}**

- The table is not in BCNF because neither EMP_DEPT nor EMP_ID alone are keys.
- To convert the given table into BCNF, we decompose it into three tables:

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

EMP_COUNTRY table:

EMP_ID	EMP_COUNTRY
264	India
264	India

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

EMP_DEPT table:

EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
Designing	D394	283
Testing	D394	300
Stores	D283	232
Developing	D283	549

EMP_DEPT_MAPPING table:

EMP_ID	EMP_DEPT
D394	283
D394	300
D283	232
D283	549

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

- **Functional dependencies:**

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

- **Candidate keys:**

For the first table: EMP_ID
For the second table: EMP_DEPT
For the third table: {EMP_ID, EMP_DEPT}

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Boyce Codd normal form (BCNF)

- **Functional dependencies:**

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

- **Candidate keys:**

For the first table: EMP_ID
For the second table: EMP_DEPT
For the third table: {EMP_ID, EMP_DEPT}

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Difference Between 1NF,2NF,3NF and BCNF

	1 NF	2 NF	3 NF	BCNF
Properties to hold	All the attributes of the relation having the atomic values. Every attribute contains single value	The table must be in 1NF, All the non-key attributes must fully functionally dependent on the Primary key of the table.	The table must be in 2NF, There is no Functional Dependency such that both Left Hand Side and Right Hand Side attributes of the FD are non-key attributes. In other words, no transitive dependency is allowed	The table must be in 3NF, For all the Functional Dependencies (FDs) hold in the table, if the FD is non-trivial then the determinant (LHS of FD) of that FD should be a Super key
What **Anomalies** exist	May allow some anomalies	May allow some anomalies	May allow some anomalies	Always eliminates anomalies
Identification of Functional Dependencies	Not necessary	Must	Must	Must
Composite Primary Key	Allowed	Allowed (if no partial dependency exists)	Allowed	Not allowed
Elimination	Eliminate repeating groups	Eliminate redundant data	Eliminate columns not dependent on key	Eliminate multiple candidate keys
Attribute Domain	Should be atomic	Should be atomic	Should be atomic	Should be atomic
Achievability	Always achievable	Always achievable	Always achievable	Not always
Lossless Join Decomposition	Always achievable	Always achievable	Always achievable	Sometimes not achievable

https://t4tutorials.com/
https://t4tutorials.com/
https://t4tutorials.com/

Closure of an Attribute Set

Closure of an Attribute Set

Closure of an Attribute can be defined as a
set of attributes that can be functionally determined from it.
Closure of a set F of FDs is the set F+ of all FDs that can be **inferred** from F

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

How to find attribute closure of an attribute set?

To find attribute closure of an attribute set:

- **Add elements of attribute set** to the result set.
- Recursively add elements to the result set which can be **functionally determined** from the elements of the result set.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Prime and non-prime attributes

Attributes which are **parts of any candidate key** of relation are called as **prime attribute**, others are **non-prime attributes.**

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

How to check whether an FD can be derived from a given FD set?

To check whether an FD A->B can be derived from an FD set F,

1. Find **(A)+** using FD set F.
2. If B is subset of (A)+, then A->B is true else not true.

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Examples

Q. In a schema with attributes A, B, C, D and E following set of functional dependencies are given

{A -> B, A -> C, CD -> E, B -> D, E -> A}

Which of the following functional dependencies is NOT implied by the above set?

A. CD -> AC

B. BD -> CD

C. BC -> CD

D. AC -> BC

The material in this presentation belongs to St. Francis Institute of Technology and is solely for educational purposes. Distribution and modifications of the content is prohibited.

Examples(Cont..)

Answer: Using FD set given in question,

(CD)+ = {CDEAB} which means CD -> AC also holds true.

(BD)+ = {BD} which means BD -> CD can't hold true. So this FD is no implied in FD set. So (B) is the required option.

Others can be checked in the same way.