Functional Dependencies

- A formal constraint among attributes that is the main tool for formally measuring the appropriateness of **attribute groupings** into relation schemas

Functional Dependencies

- Constraint between two sets of attributes from the database
- Suppose that relational database schema has ***n*** attributes $\mathbf{A_1,A_2,\ldots\ldots A_n}$
- Single universal relation schema $\mathbf{R=\{A_1,A_2,\ldots..A_n\}}$

Definition-Functional Dependency

- A functional dependency, denoted by **x→y**, between two sets of attributes x and y that are subsets of R specifies a ***constraint*** on the possible tuples that can form a relation state r of R.

- The ***constraint*** is that, for any two tuples t1 and t2 in r that have t1[x]=t2[x], they must also have t1[y]=t2[y]

Functional Dependency

- an attribute A is functionally dependent on a set of attributes X if and only if
  - value of A is determined solely by the values of X
  - values of X uniquely determine a value of A

X → A

child → mother

mother ↛ child

•The value of child implies the value of mother

Value of mother does NOT imply value of child

Child is the **determinant**

Mother is the dependent/determined

Functional Dependencies

- The values of the x component of a tuple uniquely ***determine*** the values of the y component
- y is functionally dependent on x
- x→ y (y depends upon x)

Functional Dependencies

- Consider the example of Employee Relation

EID	NAME	SALARY
1	Aditya	15000
2	Manoj	16000
3	Sandeep	9000
4	Vikas	10000
5	Manoj	9000

Functional Dependencies

- Consider the example of Employee Relation

EID	NAME	SALARY
1	Aditya	15000
2	Manoj	16000
3	Sandeep	9000
4	Vikas	10000
5	Manoj	9000

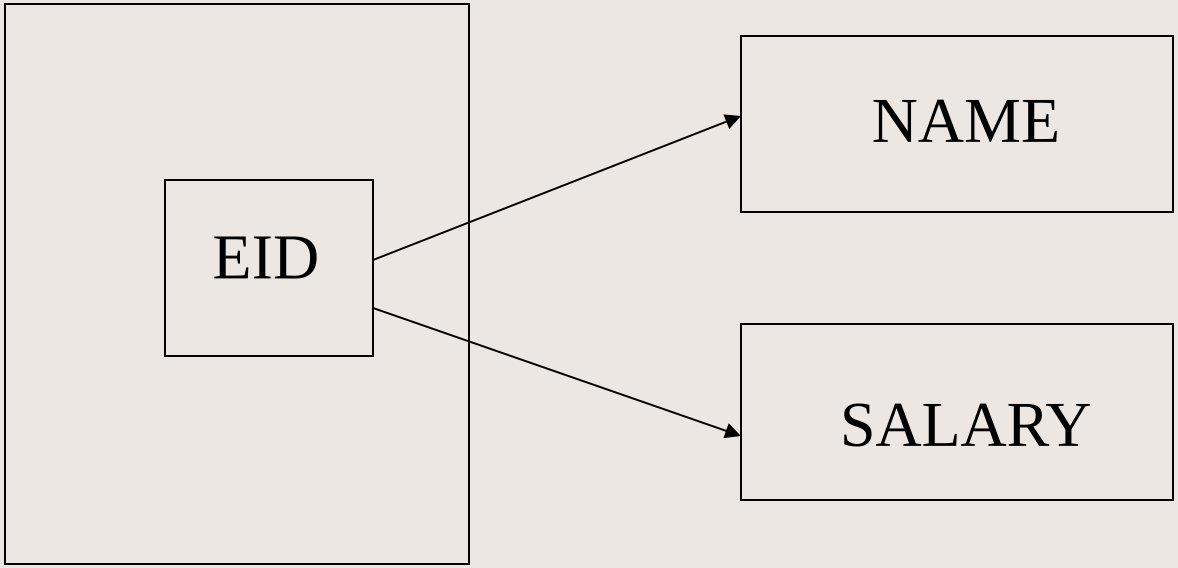
- x→ y (y depends upon x)
- EID →Name, salary
- if x (EID) has value 5 then y (Name, salary) has value Manoj, 9000

Functional Dependency

EID→Name

EID→Salary

EID→Name, Salary



Types of Functional Dependencies

- Partial Dependency & Fully Functional Dependency
- Transitive Dependency & Non Transitive Dependency
- Single valued Dependency & Multivalued Dependency
- Trival Dependency & Non -Trival Dependency

Keys

- **Superkey**
  - A superkey of a relation schema R, is a set of attributes S (subset or equal to) R with the property that no two tuples t1 & t2 in any legal relation state r of R will have t1[S]=t2[S].
- **Key**
  - A minimal superkey
- **Candidate key**
  - If the relation schema has **more than one key**, each is called a **candidate key**
  - One of the candidate key is arbitrarily designated to be the **primary key**

Prime & Nonprime Attribute

- An attribute of relation schema R is called a **prime attribute** of R if it is a **member of some candidate key of R**

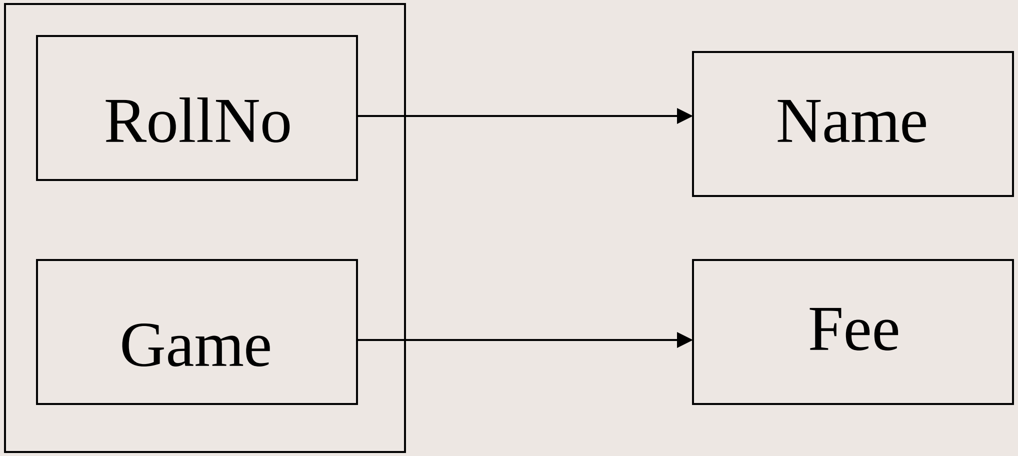
- A attribute is called **non prime** if it is **not a prime attribute**

Partial Dependency

- Suppose we have more than one attributes in primary key
- Let A be the non prime key attribute
- If A is not dependent upon all prime key attributes then partial dependency exists
- Consider schema Student(Rollno,Name,Game,Fee,Grade)

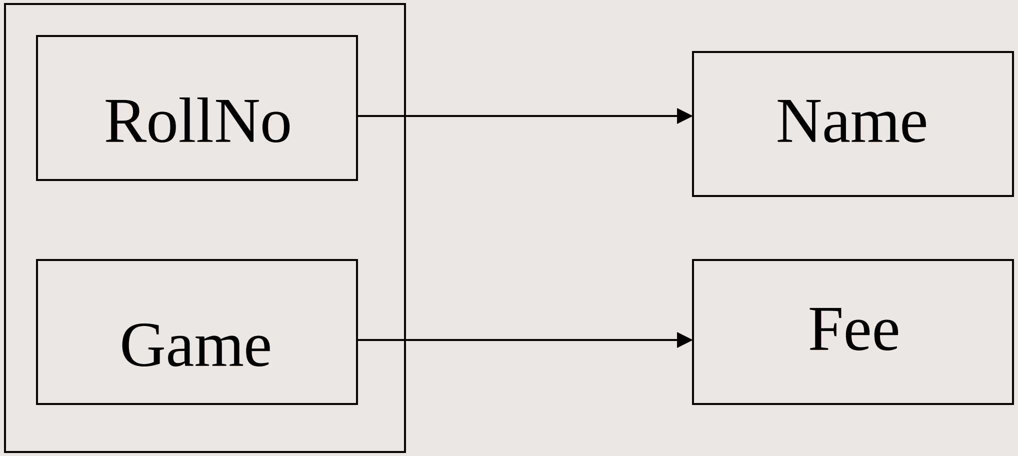
Partial Dependency

- Suppose we have more than one attributes in primary key
- Let A be the non prime key attribute
- If A is not dependent upon all prime key attributes then partial dependency exists



Partial Dependency

- Suppose we have more than one attributes in primary key
- Let A be the non prime key attribute
- If A is not dependent upon all prime key attributes then partial dependency exists



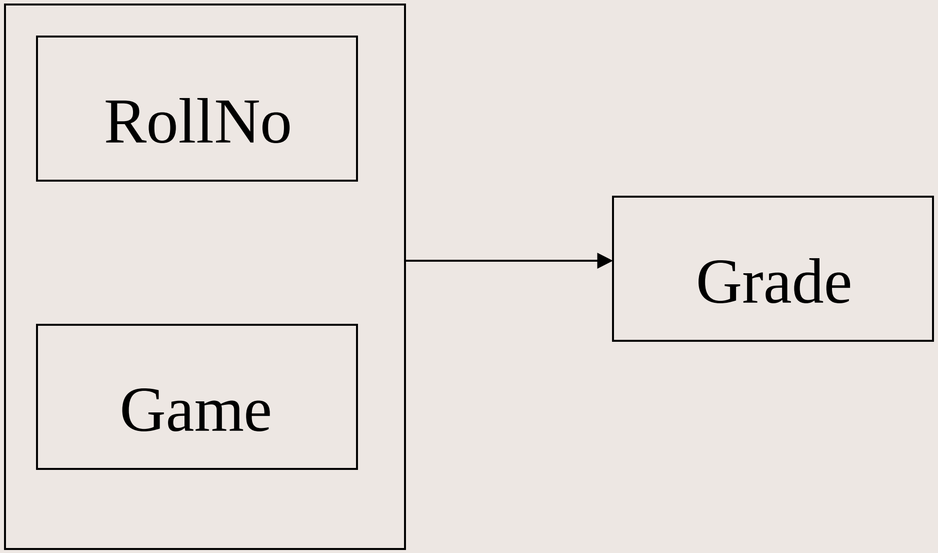
•Partial dependency is due to more than one prime key attribute

Fully Functional Dependency

- Let A be the non prime key attribute
- Value of A is dependent upon all prime key attributes
- Then A is said to be fully functional dependent

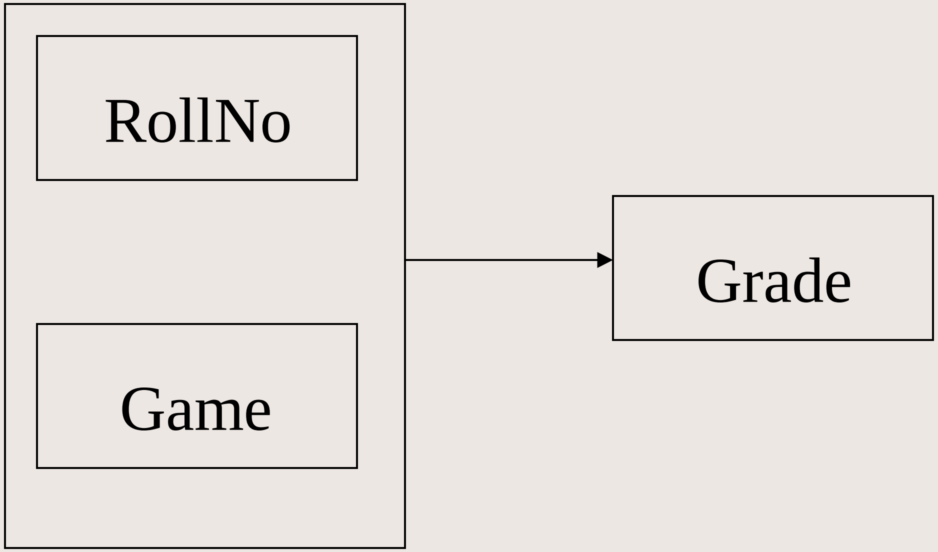
Fully Functional Dependency

- Let A be the non prime key attribute
- Value of A is dependent upon all prime key attributes
- Then A is said to be fully functional dependent



Fully Functional Dependency

- Grade is fully functionally dependent because we can find the grade of any student in a particular game if we know RollNo and Game of that student



Transitive Dependency

- Transitive dependency is due to dependency between non prime key attributes
- Suppose in a relation R

x → y

y → z

then x → z

- z is said to be transitively dependent upon x

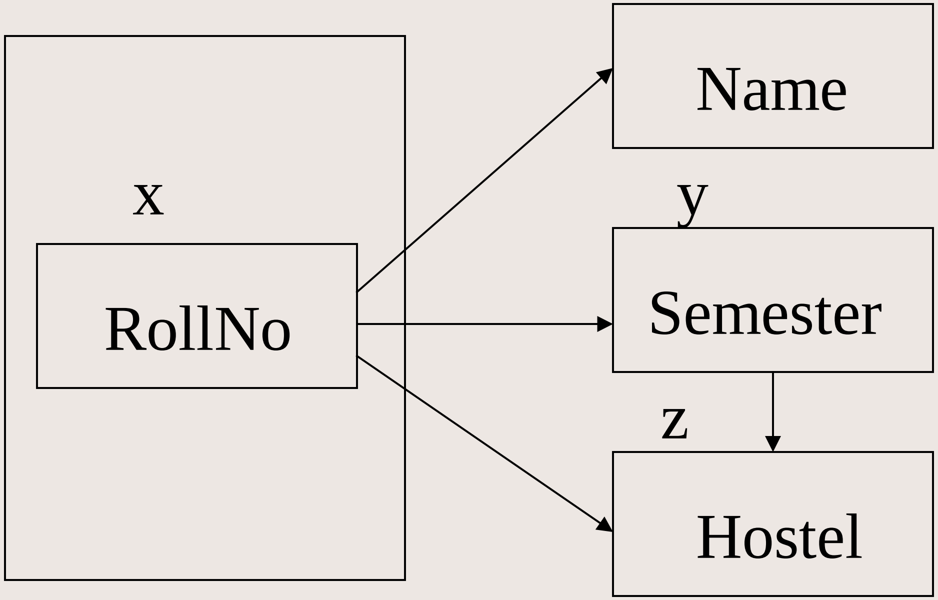
Transitive Dependency

- Consider a relation student having prime key attribute (RollNo) and non-prime key attributes (Name, Semester, Hostel)
- For each semester there is different hostel

x → y	RollNo→ Name, Semester, Hostel
y → z	**Semester→ Hostel**
x → z	RollNo→ Hostel

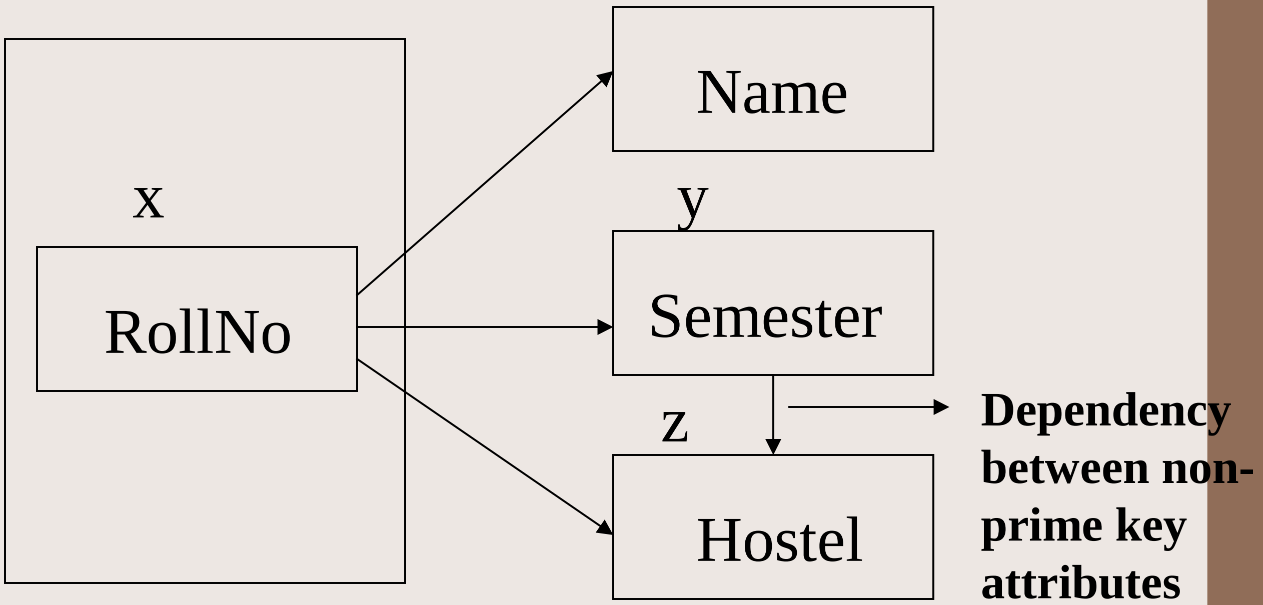
Transitive Dependency

- Consider a relation student having prime key attribute (RollNo) and non-prime key attributes (Name, Semester, Hostel)



Transitive Dependency

- Consider a relation student having prime key attribute (RollNo) and non-prime key attributes (Name, Semester, Hostel)
- Hostel is transitively dependent upon RollNo



Non-Transitive Dependency

- Any functional dependency which is not transitive
- Exists if there is no dependency between non-prime key attributes

Single valued Dependency

- In any relation R, if for a particular value of x, y has single value then it is known as single valued dependency

x → y

Multivalued Dependency

- In any relation R, if for a particular value of x, y has more than one value

x →→ y

Multivalued Dependency

- Consider the relation Teacher

ID	Teacher	Class	Days
L1	Sam	Computer	1
L2	John	Computer	6
L1	Sam	Electronics	3
L2	John	Mechanical	5
L3	Nick	Mechanical	2

Multivalued Dependency

ID	Teacher	Class	Days
L1	Sam	Computer	1
L2	John	Computer	6
L1	Sam	Electronics	3
L2	John	Mechanical	5
L3	Nick	Mechanical	2

- Teacher →→ Class
- Class →→ Days
- ID → Teacher

Trival Dependency

- In any relation R, x → y is trival if y is the subset of x

RollNo, Game → RollNo

Non - Trival Dependency

- In any relation R, x → y is non-trival if y is not the subset of x

RollNo, Game → Grade

Anomalies in Relational Database

- Anomalies refer to the undesirable results because of modification of data
  - Insertion Anomaly
  - Deletion Anomaly
  - Updation Anomaly

Insertion Anomaly

Consider the relation Employee

EID	**Name**	**Salary**	**Dept.No**	**Dept.Name**
1	Goyal	10000	2	Accounts
2	Amit	9000	2	Accounts
3	Deepak	11000	1	Sales
4	Sandeep	8500	5	Marketing
5	Vikas	7000	5	Marketing
6	Jain	15000	2	Accounts
7	Lalit	14000	5	Marketing
8	vishal	10000	2	Accounts

Insertion Anomaly

- add new information in any relation but cannot enter that data because of some constraints known as Insertion anomaly

- In relation Employee, we cannot add new department Finance unless there is an employee in Finance department

- It violates Entity Integrity Rule (Primary key cannot be NULL)

Insertion Anomaly

Employee

EID	**Name**	**Salary**	**Dept.No**	**Dept.Name**
1	Goyal	10000	2	Accounts
2	Amit	9000	2	Accounts
3	Deepak	11000	1	Sales
4	Sandeep	8500	5	Marketing
5	Vikas	7000	5	Marketing
6	Jain	15000	2	Accounts
7	Lalit	14000	5	Marketing
8	vishal	10000	2	Accounts
			10	**Finance**

Deletion Anomaly

- Occurs when we try to delete any existing information from any relation

- This causes deletion of any other undesirable information

- In relation Employee, if we try to delete tuple containing Deepak this leads to the deletion of department ‘Sales’

Deletion Anomaly

Employee

EID	Name	Salary	Dept.No	Dept.Name
1	Goyal	10000	2	Accounts
2	Amit	9000	2	Accounts
3	**Deepak**	**11000**	**1**	**Sales**
4	Sandeep	8500	5	Marketing
5	Vikas	7000	5	Marketing
6	Jain	15000	2	Accounts
7	Lalit	14000	5	Marketing
8	vishal	10000	2	Accounts

Updation Anomaly

- Occurs when we try to update any existing information in any relation

- This causes inconsistency of data

- In relation Employee, if we try to change Dept.No of department **Accounts** for a tuple

Updation Anomaly

Employee

EID	**Name**	**Salary**	**Dept.No**	**Dept.Name**
1	Goyal	10000	2	Accounts
2	Amit	9000	2	Accounts
3	Deepak	11000	1	Sales
4	Sandeep	8500	5	Marketing
5	Vikas	7000	5	Marketing
6	Jain	15000	2	Accounts
7	Lalit	14000	5	Marketing
8	**vishal**	**10000**	**2**	**Accounts**

Updation Anomaly

- This causes inconsistency of data

- Search all employees working in Accounts department and update them individually

NORMALIZATION

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce-Codd Normal Form

First Normal Form (1NF)

- It states that the domain of an attribute must include only **atomic** values

- The value of any attribute in a tuple must be a **single value** from the domain of that attribute

First Normal Form (1NF)

Consider the relation Employee

EID	**Name**	**Salary**	**Dept.No**	**Dept.Name**
1	Shivi Goyal	10000	2	Accounts
2	Amit Chopra	9000	2	Accounts
3	Deepak Gupta	11000	1	Sales
4	Sandeep Sharma	8500	5	Marketing
5	Vikas Malik	7000	5	Marketing
6	Gaurav Jain	15000	2	Accounts
7	Lalit Parmar	14000	5	Marketing
8	Vishal Bamel	10000	2	Accounts

First Normal Form (1NF)

Employee

<u>EID</u>	Name	Salary	Dept.No	Dept.Name
1	Shivi Goyal	10000	2	Accounts
2	Amit Chopra	9000	2	Accounts
3	Deepak Gupta	11000	1	Sales
4	Sandeep Sharma	8500	5	Marketing
5	Vikas Malik	7000	5	Marketing
6	Gaurav Jain	15000	2	Accounts
7	Lalit Parmar	14000	5	Marketing
8	Vishal Bamel	10000	2	Accounts

- **Name is not atomic**

First Normal Form (1NF)

Relation Employee is in 1NF

EID	First-Name	Last-Name	Salary	Dept.No	Dept.Name
1	Shivi	Goyal	10000	2	Accounts
2	Amit	Chopra	9000	2	Accounts
3	Deepak	Gupta	11000	1	Sales
4	Sandeep	Sharma	8500	5	Marketing
5	Vikas	Malik	7000	5	Marketing
6	Gaurav	Jain	15000	2	Accounts
7	Lalit	Parmar	14000	5	Marketing
8	Vishal	Bamel	10000	2	Accounts

- **Anomalies described earlier are also applicable here**

Second Normal Form (2NF)

- A relation schema R is in 2NF if every nonprime attribute A in R is ***fully functionally dependent*** on the primary key of R

Second Normal Form (2NF)

Consider the relation Student

RollNo	**Game**	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

Second Normal Form (2NF)

Consider the relation **Student**

RollNo	**Game**	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

RollNo→Name
Game→Fee
RollNo,Game→Grade

Second Normal Form (2NF)

<u>**RollNo**</u>	<u>**Game**</u>	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

- Relation Student is in 1NF but still contains anomalies
  - Insertion Anomaly: Want to add a new game
  - Deletion Anomaly: Delete student Jack;
    loose information about Hockey
  - Updation Anomaly: Change Fee of Cricket

Second Normal Form (2NF)

RollNo	**Game**	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

- **Solution?????**

Second Normal Form (2NF)

RollNo	**Game**	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

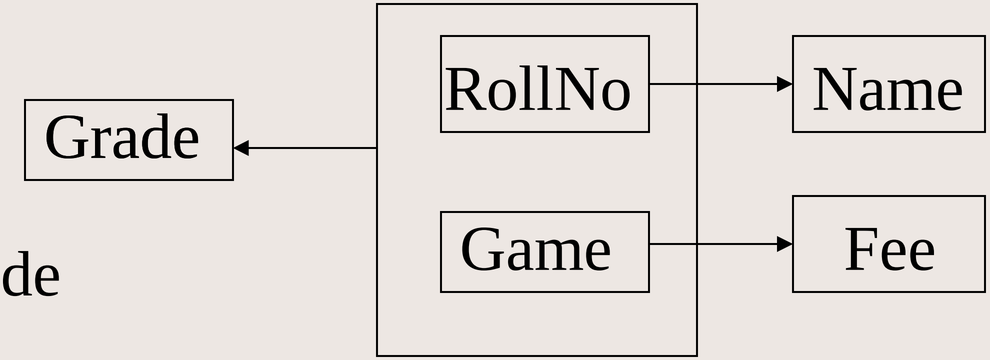
- **Solution**
  - Separate Partial dependencies and Fully functional dependencies

Second Normal Form (2NF)

Consider the relation **Student**

RollNo	**Game**	**Name**	**Fee**	**Grade**
CSU1	Cricket	Amit	200	A
CSU2	Badminton	Deepak	150	B
CSU3	Cricket	Sandeep	200	A
CSU4	Badminton	Vikas	150	C
CSU5	Hockey	Jack	100	A
CSU6	Cricket	John	200	C

RollNo→Name
Game→Fee
RollNo,Game→Grade



Second Normal Form (2NF)

Student (RollNo,Game,Name,Fee,Grade)

RollNo→Name

Game→Fee

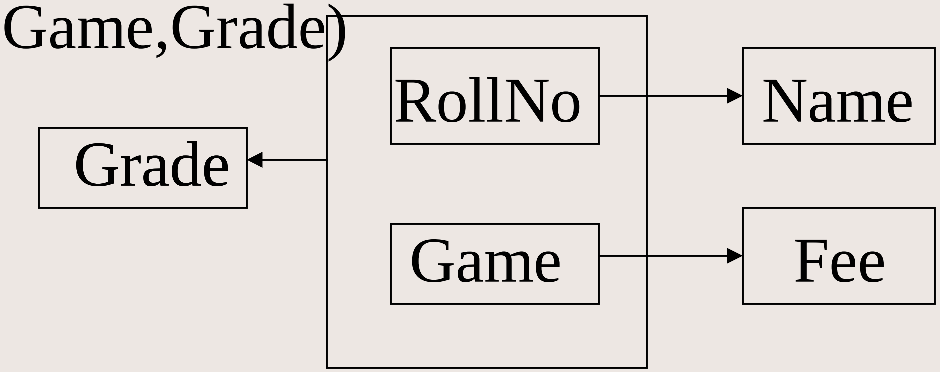
RollNo,Game→Grade

- Into three relations

Student (RollNo,Name)

Games (Game,Fee)

Performance (RollNo,Game,Grade)



Second Normal Form (2NF)

Student

RollNo	**Name**
CSU1	Amit
CSU2	Deepak
CSU3	Sandeep
CSU4	Vikas
CSU5	Jack
CSU6	John

Games

Game	Fee
Cricket	200
Badminton	150
Hockey	100

Performance

RollNo	**Game**	**Grade**
CSU1	Cricket	A
CSU2	Badminton	B
CSU3	Cricket	A
CSU4	Badminton	C
CSU5	Hockey	A
CSU6	Cricket	C

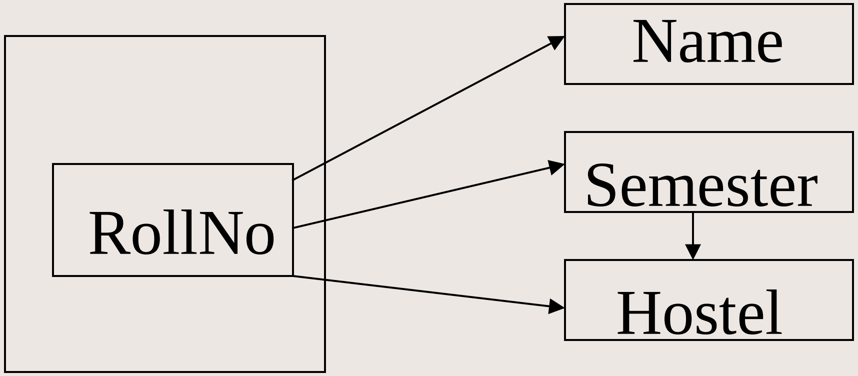
Third Normal Form (3NF)

- A relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key

Third Normal Form (3NF)

Consider the relation Student

<u>RollNo</u>	**Name**	**Semester**	**Hostel**
CSU1	Amit	1	H1
CSU2	Deepak	2	H2
CSU3	Sandeep	1	H1
CSU4	Vikas	4	H4
CSU5	Jack	3	H3



Third Normal Form (3NF)

Consider the relation Student

RollNo	Name	Semester	Hostel
CSU1	Amit	1	H1
CSU2	Deepak	2	H2
CSU3	Sandeep	1	H1
CSU4	Vikas	4	H4
CSU5	Jack	3	H3

- Primary key: RollNo
- Different hostel is allotted for different semester
- Student relation is in 2NF but still contains anomalies

Third Normal Form (3NF)

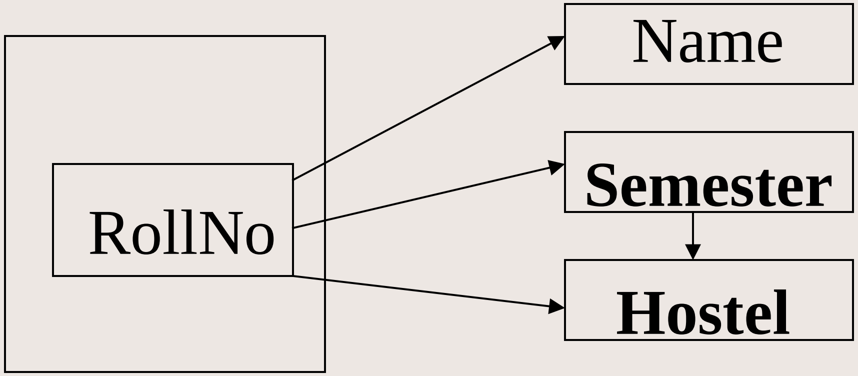
<u>RollNo</u>	**Name**	**Semester**	**Hostel**
CSU1	Amit	1	H1
CSU2	Deepak	2	H2
CSU3	Sandeep	1	H1
CSU4	Vikas	4	H4
CSU5	Jack	3	H3

- Insertion anomaly: Cannot add a new hostel - not allotted to any student
- Deletion anomaly: Cannot delete student Deepak - loose information about hostel H2
- Updation anomaly: Change hostel of first semester

Third Normal Form (3NF)

RollNo	**Name**	**Semester**	**Hostel**
CSU1	Amit	1	H1
CSU2	Deepak	2	H2
CSU3	Sandeep	1	H1
CSU4	Vikas	4	H4
CSU5	Jack	3	H3

Student (RollNo, Name, **Semester, Hostel**)



Third Normal Form (3NF)

RollNo	**Name**	**Semester**	**Hostel**
CSU1	Amit	1	H1
CSU2	Deepak	2	H2
CSU3	Sandeep	1	H1
CSU4	Vikas	4	H4
CSU5	Jack	3	H3

- **Solution?????**

Third Normal Form (3NF)

Student (RollNo, Name, Semester)
Hostels (Semester, Hostel)

Student

RollNo	Name	Semester
CSU1	Amit	1
CSU2	Deepak	2
CSU3	Sandeep	1
CSU4	Vikas	4
CSU5	Jack	3

Hostels

Semester	Hostel
1	H1
2	H2
3	H3
4	H4

Boyce Codd Normal Form (BCNF)

- A relation schema R is in BCNF if whenever a nontrivial functional dependency **x**→A holds in R, then **x is a superkey of R**

Boyce Codd Normal Form (BCNF)

- BCNF is a strict format of 3NF

- A relation schema R is in BCNF if and only if all determinants are super keys

- BCNF deals with multiple candidate keys

Boyce Codd Normal Form (BCNF)

Consider the relation Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6

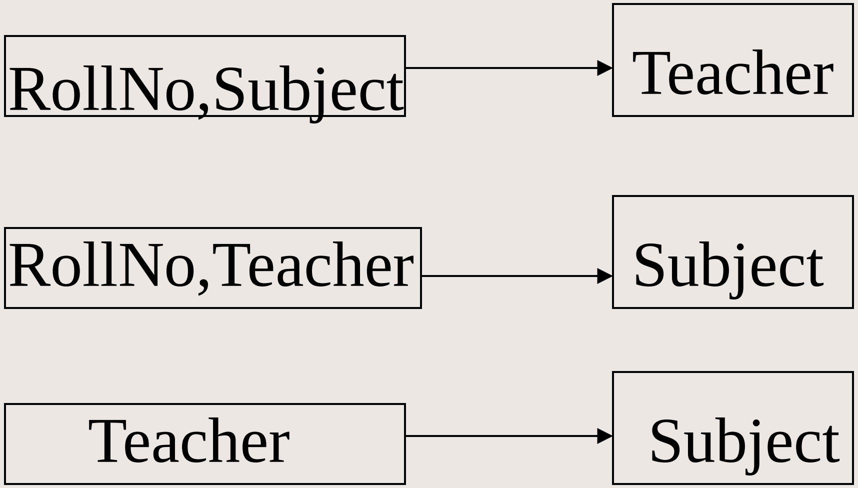
Assumptions:

- Student can have more than 1 subject
- A Teacher can teach only 1 subject
- A subject can be taught by more than 1 teacher

Boyce Codd Normal Form (BCNF)

Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6



RollNo,Subject → Teacher

RollNo,Teacher → Subject

Teacher → Subject

Boyce Codd Normal Form (BCNF)

Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	**VB**	**T4**
8	Oracle	T6

- Deletion Anomaly - if we delete student with RollNo.7; loose information about Teacher T4

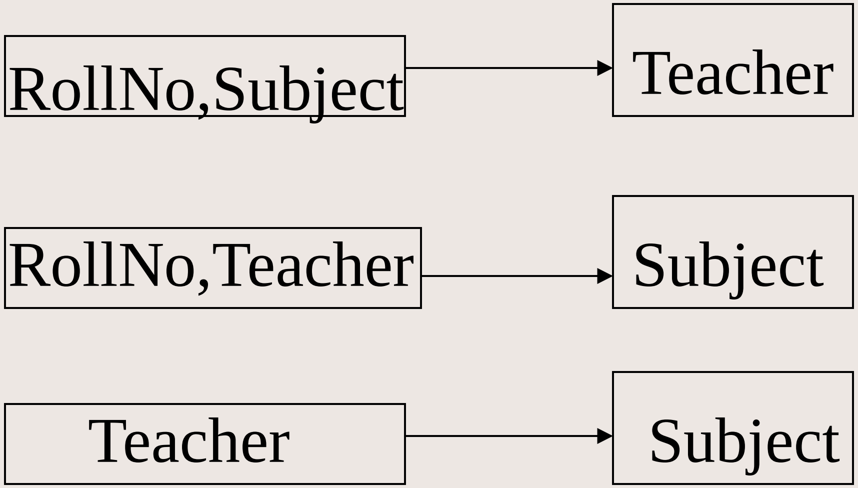
- Insertion Anomaly - If we want to add new subject VC++

- Updation Anomaly - want to change Teacher for subject C.

Boyce Codd Normal Form (BCNF)

Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6



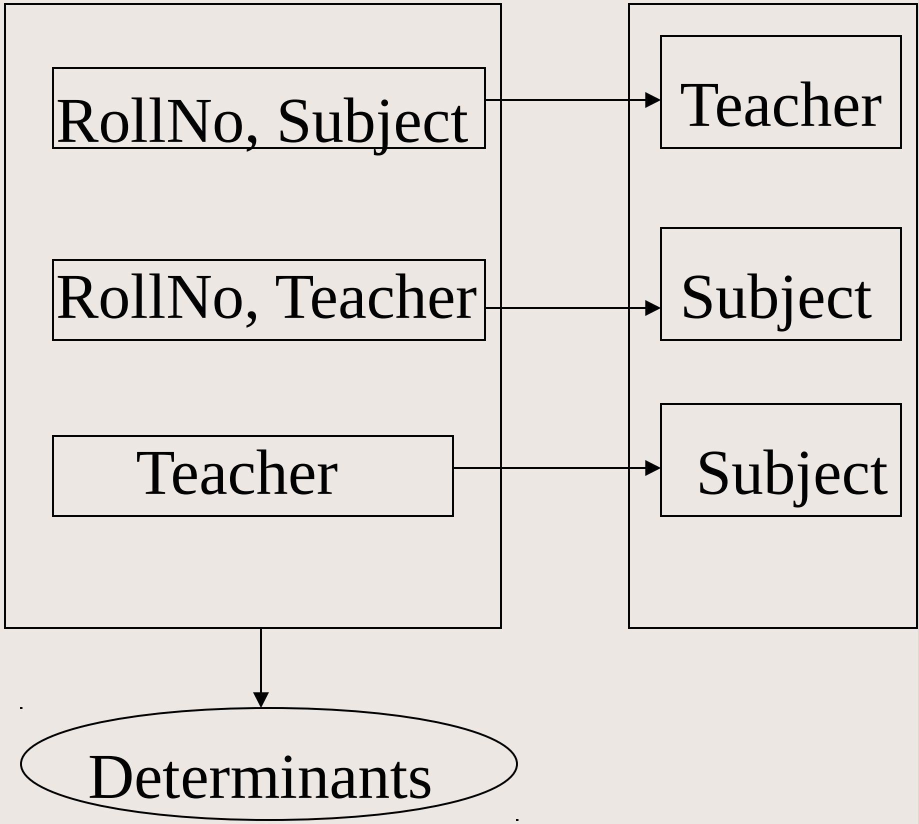
Two candidate keys

- (RollNo, Subject)
- (RollNo, Teacher)

Boyce Codd Normal Form (BCNF)

Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6



RollNo, Subject → Teacher

RollNo, Teacher → Subject

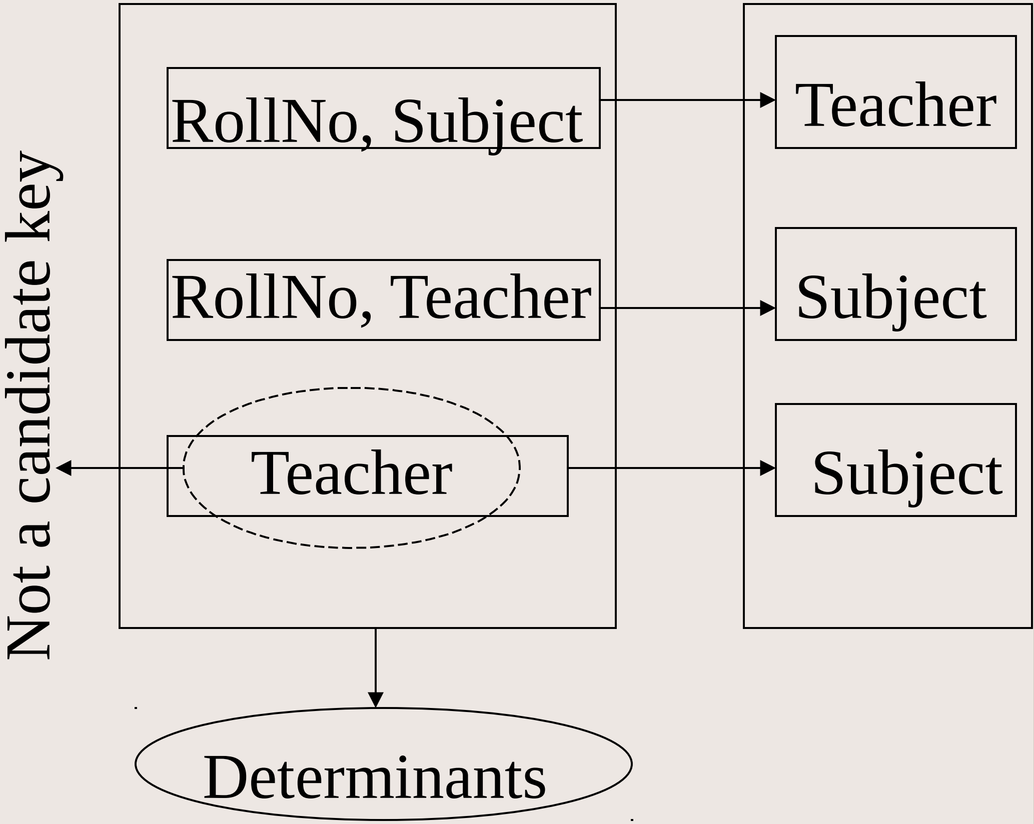
Teacher → Subject

Determinants

Boyce Codd Normal Form (BCNF)

Student

RollNo	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6



Boyce Codd Normal Form (BCNF)

- **Solution**
- Student-Teacher (RollNo, Teacher)
- Teacher-subject (Teacher, subject)

Student-Teacher

RollNo	Teacher
1	T1
2	T2
3	T1
4	T3
5	T3
1	T5
6	T5
3	T2
7	T4
8	T6

Teacher-subject

Teacher	Subject
T1	C
T2	C++
T3	Java
T4	VB
T5	Oracle
T6	Oracle

THANK YOU