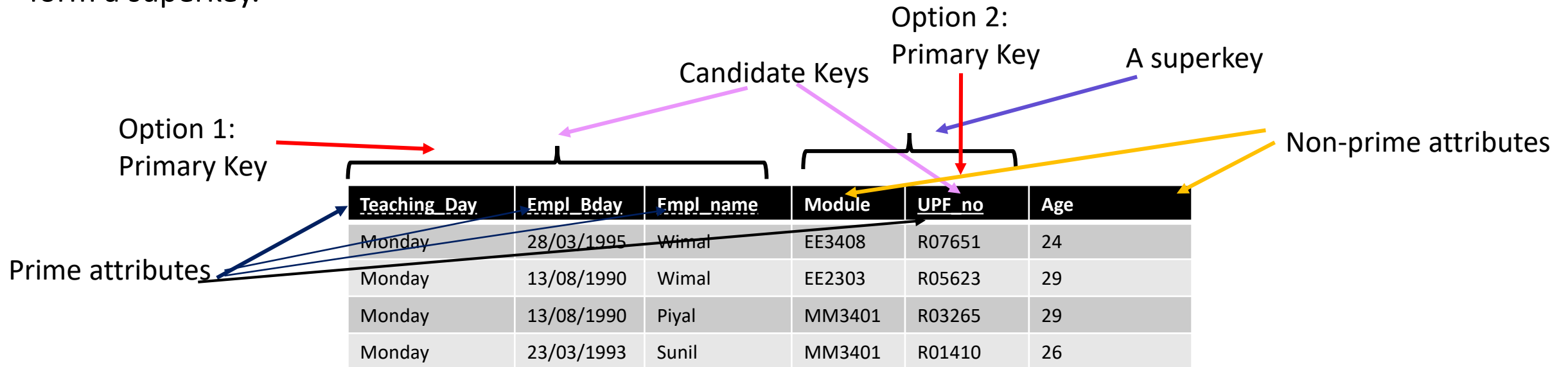


EE4202 Database Systems

DATA DEPENDENCIES


ATTRIBUTES IN RELATIONS

- A relation schema may have more than one key. Each of the keys is called a **candidate key**.
- Any attribute that is part of a candidate key of a relation is also known as **prime attribute** (Partial key in ER diagrams). All other non-candidate key attributes are known as non-prime attributes.
- One of the candidate keys is designated as the **primary key**. All other keys are secondary keys which are designated as **unique keys**.
- A **superkey** is a set of attributes which must have distinct tuples for the set of attributes.
- A **key/candidate key** is a *minimal superkey*. Removal of any attribute will not cause remaining attributes to form a superkey.



FUNCTIONAL DEPENDENCY

- Schema based constraints can be defined using a Data Definition Language (DDL). Ex:SQL
- **Functional dependency** between two sets of attributes X and Y of a relation exists if a value of X determines a unique value of Y for all relation instance (extension).
- Notation is $X \longrightarrow Y$. In schema, X is notated by vertical lines and Y by arrows in a connected horizontal line.
- Functional dependency for any subset Y of a relation always exist if X is a candidate key of the relation.




A diagram above the table shows a horizontal line with four vertical arrows pointing down to the Department, Area, Designation, and Field_Supervisor columns, indicating functional dependencies.

<u>Employee_ID</u>	Department	Area	Designation	Field_Supervisor
EE001	EIE	Software	Lecturer	NULL
EE002	EIE	Communication	Senior Lecturer	EE001
MM007	MME	Automobile	Lecturer	NULL
MM001	MME	Manufacturing	Senior Lecturer	NULL
CE001	CEE	Hydraulic	Senior Lecturer	NULL
CE003	CEE	Structural	Lecturer	CE001
CE007	CEE	Environment	Lecturer	CE001
EE005	EIE	Electronics	Lecturer	EE001

- Ex: Each Department, Area, Designation, Field_Supervisor are functionally dependant on Employee_ID as it is the primary key.
- $\text{Employee_ID} \longrightarrow \{\text{Department, Area, Designation, Field_supervisor}\}$
- Area is not functionally dependant on Department. As for EIE it can have Software, Communication, Electronics etc so that a value of Department does not uniquely determine a value of Area.

FUNCTIONAL DEPENDENCY

- **Full functional dependency** $X \twoheadrightarrow Y$ exists if an attribute is removed from X , the functional dependency is removed. Ex: $\text{Employee_ID} \twoheadrightarrow \text{Emp_Name}$
- **Partial functional dependency** $X \twoheadrightarrow Y$ exist if an attribute is removed from X , some of the functional dependencies still hold.
- Ex: $\{\text{Employee_ID}, \text{Module}\} \twoheadrightarrow \{\text{Emp_Name}\}$ is a partial functional dependency as removal of attribute Module will cause $\{\text{Employee_ID}\} \twoheadrightarrow \{\text{Emp_Name}\}$ which is still a functional dependency.



<u>Employee_ID</u>	<u>Module</u>	Emp_Name	Teaching_time
EE001	EE3408	Wimal	56
EE002	EE2303	Piyal	42
MM007	MM3401	Nimal	56
MM001	MM3482	Sunil	56
CE001	CE1201	Amara	28
CE003	CE2281	Nayana	28
CE007	CE3103	Nalini	14
EE005	EE6206	Upali	28

TRANSITIVE DEPENDENCY

- Transitive dependency is a special type of functional dependency.
- A functional dependency $X \longrightarrow Y$ is said to be transitive dependent if there exist a set of attributes Z ; where Z is neither a candidate key nor a subset of any key that satisfy both;
 $X \longrightarrow Z$ and $Z \longrightarrow Y$.

EMPLOYEE

<u>Employee_ID</u>	Department_ID	Area	Designation	Dept_Head
EE001	EIE	Software	Lecturer	EE001
EE002	EIE	Communication	Senior Lecturer	EE001
MM007	MME	Automobile	Lecturer	MM003
MM001	MME	Manufacturing	Senior Lecturer	MM003
CE001	CEE	Hydraulic	Senior Lecturer	CE001
CE003	CEE	Structural	Lecturer	CE001
CE007	CEE	Environment	Lecturer	CE001
EE005	EIE	Electronics	Lecturer	EE001

- Ex: $\text{Employee_ID} \longrightarrow \text{Dept_Head}$ is a transitive dependency which is transitive through Department_ID
- The functional dependencies,
 $\text{Employee_ID} \longrightarrow \text{Department_ID}$ and
 $\text{Department_ID} \longrightarrow \text{Dept_Head}$ exist.
- None of Department_ID nor Dept_Head is a prime attribute in EMPLOYEE relation.

TRANSITIVE DEPENDENCY-EXAMPLE

AUTHORS

Author_ID	Author_Name	Book	Author_Nationality
Auth_001	Orson Scott Card	Ender's Game	United States
Auth_001	Orson Scott Card	Children of the Mind	United States
Auth_002	Margaret Atwood	The Handmaid's Tale	Canada
Auth_003	Richard Bach	Stranger to the ground	United States

- Ex: $\text{Book} \longrightarrow \text{Author_Nationality}$ is a transitive dependency which is transitive through Author_Name
- The functional dependencies,
 - $\text{Book} \longrightarrow \text{Author_Name}$ and
 - $\text{Author_Name} \longrightarrow \text{Author_Nationality}$
- Then this table introduces a transitive dependency:
 $\text{Book} \longrightarrow \text{Author_Nationality}$
- None of **Author_Name** nor **Author_Nationality** is a prime attribute in AUTHORS relation.

MULTIVALUED DEPENDENCY

- This is due to first normal form not allowing multivalued attributes.
- An employee Silva works on two projects: P1 and P2 and have 2 dependents: Sunil and Kamal.
- Employee's projects and dependents are independent of one another.

<u>Emp_name</u>	<u>Proj_name</u>	<u>Dep_name</u>
Silva	P1	Sunil
Silva	P2	Kamal
Silva	P1	Kamal
Silva	P2	Sunil

- Proj_name and Dep_name are independent of each other.
- In this case, we say that Emp_name multidetermines Proj_name and Dep_name.
- These dependencies can be represented as,

Emp_name \twoheadrightarrow Proj_name

Emp_name \twoheadrightarrow Dep_name

- Emp_name \twoheadrightarrow {Proj_name, Dep_name} is a multivalued dependency

MULTIVALUED DEPENDENCY

- **Trivial Multivalued dependency**

- $x \twoheadrightarrow y$ is a trivial MVD if
 - a) Y is a subset of X ($Y \subseteq X$) or
 - b) $X \cup Y = R$


☐ $\text{Emp_name} \twoheadrightarrow \text{Proj_name}$

☐ $\text{Emp_name} \twoheadrightarrow \text{Dep_name}$


- **Non-Trivial Multivalued dependency**

- An MVD that satisfies neither (a) nor (b) is called a nontrivial MVD.

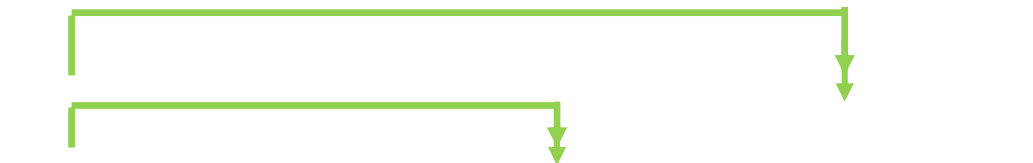
☐ $\text{Emp_name} \twoheadrightarrow \text{Proj_name}$



<u>Emp_name</u>	<u>Proj_name</u>
Silva	P1
Silva	P2



<u>Emp_name</u>	<u>Dep_name</u>
Silva	Sunil
Silva	Kamal



<u>Emp_name</u>	<u>Proj_name</u>	<u>Dep_name</u>
Silva	P1	Sunil
Silva	P2	Kamal
Silva	P1	Kamal
Silva	P2	Sunil