Database Management Systems (CSN-351)
Relational Database Design

BTech 3rd Year (CS) + Minor + Audit

Instructor: **R**anita **B**iswas
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand - 247 667, India

Design Alternative: Larger Schemas

instructor(<u>ID</u>, *name*, *dept_name*, *salary*)

department(<u>*dept_name*</u>, *building*, *budget*)

Design Alternative: Larger Schemas

instructor(<u>ID</u>, *name*, *dept_name*, *salary*)

department(<u>*dept_name*</u>, *building*, *budget*)

inst_dept (ID, *name*, *salary*, *dept_name*, *building*, *budget*)

Design Alternative: Larger Schemas

instructor(<u>ID</u>, *name*, *dept_name*, *salary*)

department(<u>*dept_name*</u>, *building*, *budget*)

inst_dept (ID, *name*, *salary*, *dept_name*, *building*, *budget*)

- Redundancy — inconsistency of information

Design Alternative: Larger Schemas

instructor(<u>ID</u>, *name*, *dept_name*, *salary*)

department(<u>*dept_name*</u>, *building*, *budget*)

inst_dept (ID, *name*, *salary*, *dept_name*, *building*, *budget*)

- Redundancy — inconsistency of information
- Creation of new department with no instructor

Design Alternative: Smaller Schemas

inst_dept (*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

Design Alternative: Smaller Schemas

inst_dept (*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

- Discovering repetition

Design Alternative: Smaller Schemas

inst_dept (*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

- Discovering repetition
- Each specific value for *dept_name* corresponds to at most one *budget*.

Design Alternative: Smaller Schemas

inst_dept (*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

- Discovering repetition
- Each specific value for *dept_name* corresponds to at most one *budget*.
- If there were a schema (*dept_name*, *budget*), then *dept_name* is able to serve as the primary key.

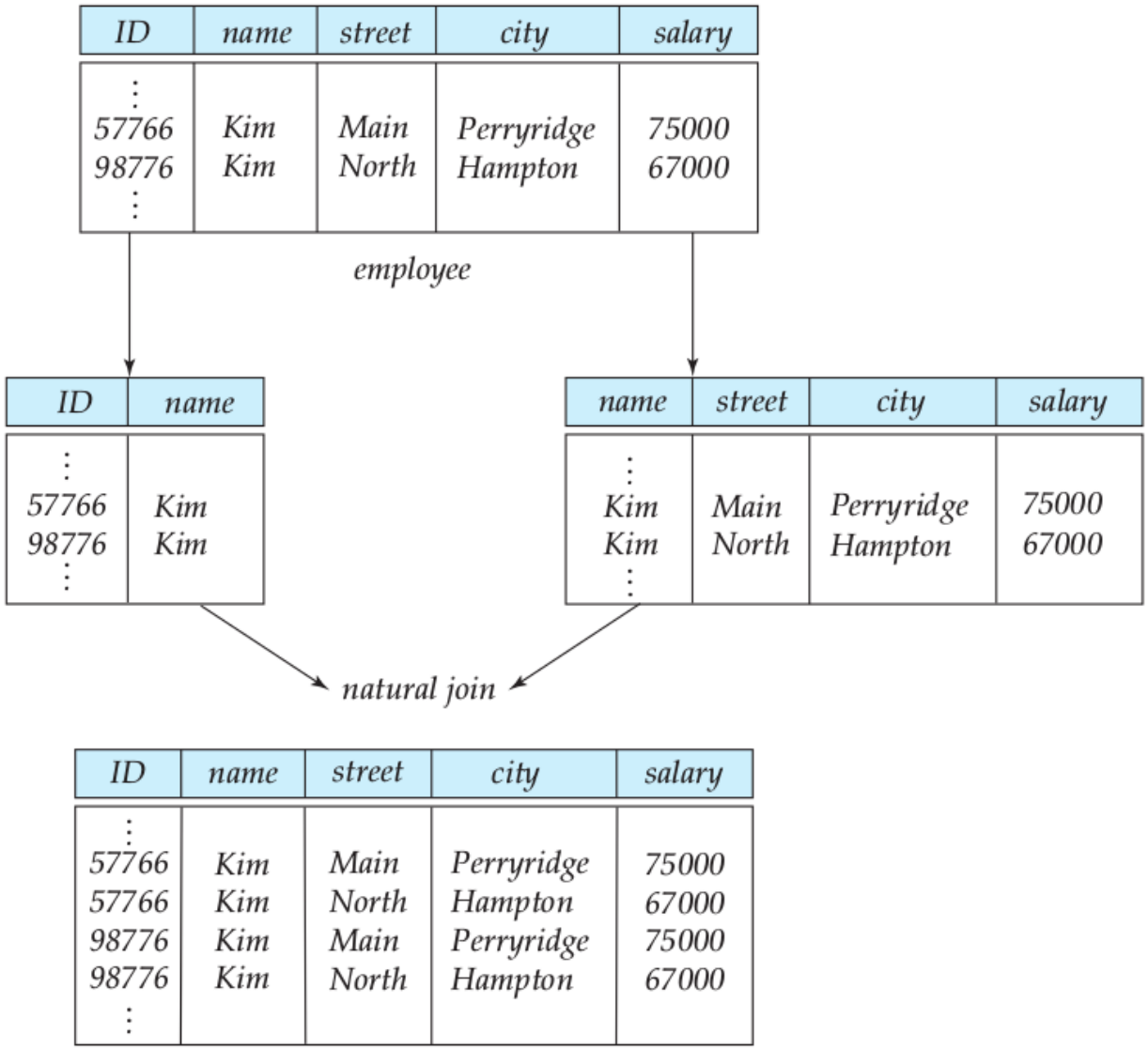
Design Alternative: Smaller Schemas

inst_dept (*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

- Discovering repetition
- Each specific value for *dept_name* corresponds to at most one *budget*.
- If there were a schema (*dept_name*, *budget*), then *dept_name* is able to serve as the primary key.

Functional Dependency: $dept\_name \rightarrow budget$

Bad Decomposition



1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.

1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.
- A relation schema R is in *first normal form (1NF)* if the domains of all attributes of R are atomic.

1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.
- A relation schema R is in *first normal form (1NF)* if the domains of all attributes of R are atomic.
- Multivalued and Composite attributes

1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.
- A relation schema R is in *first normal form (1NF)* if the domains of all attributes of R are atomic.
- Multivalued and Composite attributes
- How we use domain elements in our database?

1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.
- A relation schema R is in *first normal form (1NF)* if the domains of all attributes of R are atomic.
- Multivalued and Composite attributes
- How we use domain elements in our database?
- Use of set-valued attributes can lead to designs with redundancy.

1NF

- A domain is *atomic* if elements of the domain are considered to be indivisible units.
- A relation schema R is in *first normal form (1NF)* if the domains of all attributes of R are atomic.
- Multivalued and Composite attributes
- How we use domain elements in our database?
- Use of set-valued attributes can lead to designs with redundancy.
- Modern database systems do support many types of nonatomic values.

Constraints of the University Database

- Students and instructors are uniquely identified by their ID.
- Each student and instructor has only one name.
- Each instructor and student is (primarily) associated with only one department.
- Each department has only one value for its budget, and only one associated building.

Constraints of the University Database

- Students and instructors are uniquely identified by their ID.
- Each student and instructor has only one name.
- Each instructor and student is (primarily) associated with only one department.
- Each department has only one value for its budget, and only one associated building.

Legal instance

Keys and FDs

Definition (Superkey)

Let $r(R)$ be a relation schema. A subset K of R is a superkey of $r(R)$ if, in any legal instance of $r(R)$, for all pairs t_1 and t_2 of tuples in the instance of r if $t_1 \neq t_2$, then $t_1[K] \neq t_2[K]$.

Keys and FDs

Definition (Superkey)

Let $r(R)$ be a relation schema. A subset K of R is a superkey of $r(R)$ if, in any legal instance of $r(R)$, for all pairs t_1 and t_2 of tuples in the instance of r if $t_1 \neq t_2$, then $t_1[K] \neq t_2[K]$.

- Superkey is a set of attributes that uniquely identifies an entire tuple.

Keys and FDs

Definition (Superkey)

Let $r(R)$ be a relation schema. A subset K of R is a superkey of $r(R)$ if, in any legal instance of $r(R)$, for all pairs t_1 and t_2 of tuples in the instance of r if $t_1 \neq t_2$, then $t_1[K] \neq t_2[K]$.

- Superkey is a set of attributes that uniquely identifies an entire tuple.
- A functional dependency allows us to express constraints that uniquely identify the values of certain attributes.

Keys and FDs (contd.)

Definition (Functional Dependency)

Consider a relation schema $r(R)$, and let $\alpha \subseteq R$ and $\beta \subseteq R$.

- Given an instance of $r(R)$, we say that the instance **satisfies** the functional dependency $\alpha \rightarrow \beta$ if for all pairs of tuples t_1 and t_2 in the instance such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.

- We say that the functional dependency $\alpha \rightarrow \beta$ **holds** on schema $r(R)$ if, in every legal instance of $r(R)$ it satisfies the functional dependency.

Which FDs are satisfied?

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

Which FDs are satisfied?

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

- A functional dependency of the form $\alpha \rightarrow \beta$ is *trivial* if $\beta \subseteq \alpha$

Which FDs are satisfied?

building	*room_number*	*capacity*
Packard	101	500
Painter	514	10
Taylor	3128	70
Watson	100	30
Watson	120	50

Question 1

Consider a relational table with a single record for each registered student with the following attributes.

- *Registration_Num*: Unique registration number of each registered student
- *UID*: Unique identity number, unique at the national level for each citizen
- *BankAccount_Num*: Unique account number at the bank. A student can have multiple accounts or join accounts. This attribute stores the primary account number.
- *Name*: Name of the student
- *Hostel_Room*: Room number of the hostel

Which one of the following option is INCORRECT?

- *Registration_Num* can be a primary key
- *UID* is candidate key if all students are from the same country
- If S is a superkey such that $S \cap UID$ is $NULL$ then $S \cup UID$ is also a superkey
- *BankAccount_Num* is candidate key

Question 2

Consider the relation schema $R = \{E, F, G, H, I, J, K, L, M, N\}$ and the set of functional dependencies
$\{\{E, F\} \to \{G\}, \{F\} \to \{I, J\}, \{E, H\} \to \{K, L\}, \{K\} \to \{M\}, \{L\} \to \{N\}\}$ on R.
What is a candidate key for R?

- $\{E, F\}$
- $\{E, F, H\}$
- $\{E, F, H, K, L\}$
- $\{E\}$