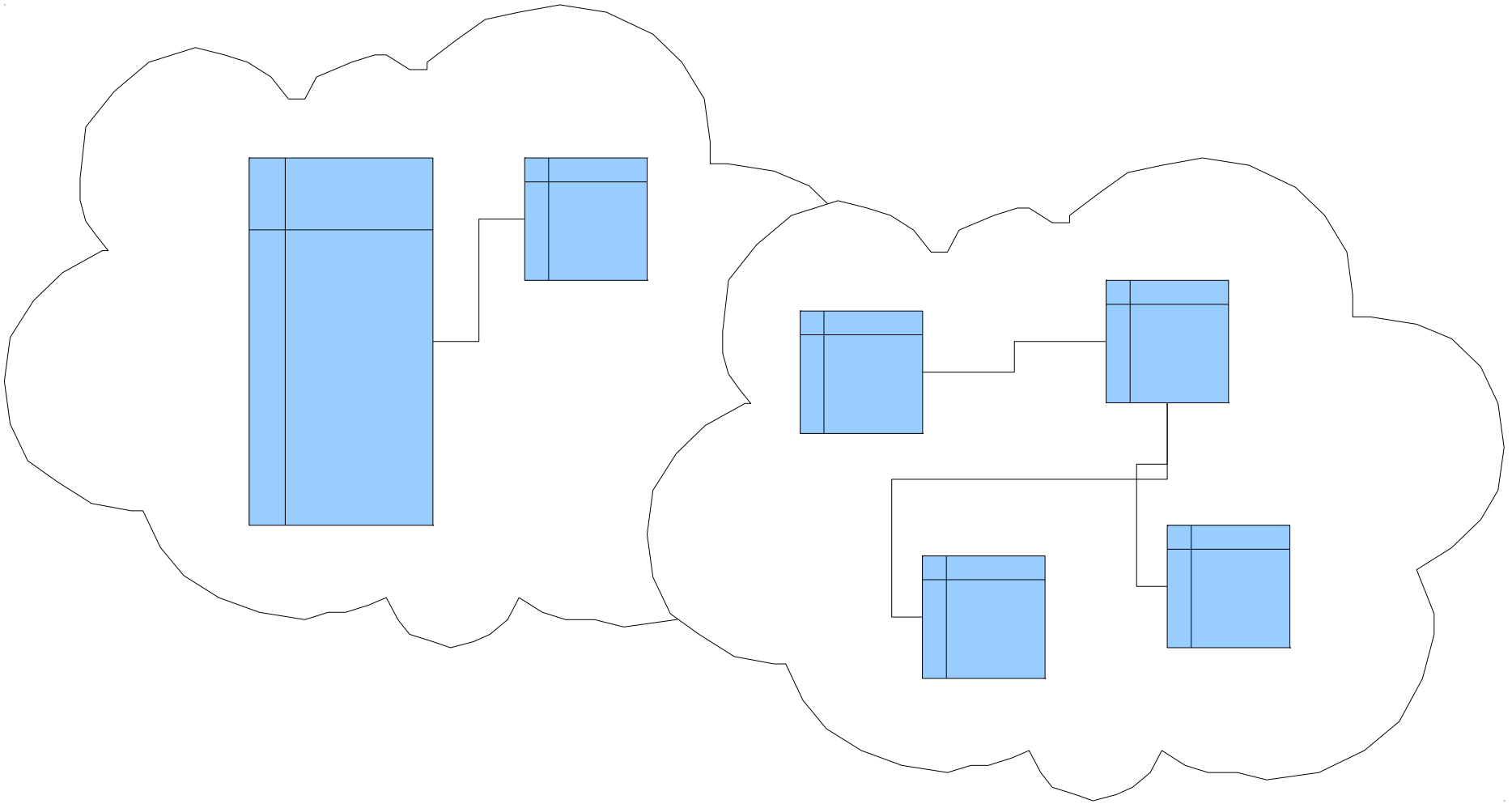
Lection 5

Normal forms

Subject area



Levels of modeling

Subject area

Subject area model

Logical data model

Physical data model

Database and application

Normal forms

The normal forms (NF) of relational database theory provide criteria for determining a table's degree of vulnerability to logical inconsistencies and anomalies. The higher the normal form applicable to a table, the less vulnerable it is to inconsistencies and anomalies. Each table must be in a "highest normal form" (HNF).

The normal forms are applicable to individual tables; to say that an entire database is in normal form n is to say that all of its tables are in normal form n.

Newcomers to database design sometimes suppose that normalization proceeds in an iterative fashion, i.e. a 1NF design is first normalized to 2NF, then to 3NF, and so on. Achieving the "higher" normal forms (above 3NF) does not usually require an extra expenditure of effort on the part of the designer, because 3NF tables usually need no modification to meet the requirements of these higher normal forms.

1NF

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
789	Maria	Fernandez	555-808-9633

The designer then becomes aware of a requirement to record **multiple** telephone numbers for some customers. He reasons that the simplest way of doing this is to allow the "Telephone Number" field in any given record to contain more than one value:

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659 555-776-4100
789	Maria	Fernandez	555-808-9633

1NF

The designer might attempt to get around this restriction by defining multiple Telephone Number columns:

Customer

Customer ID	First Name	Surname	Tel. No. 1	Tel. No. 2	Tel. No. 3
123	Robert	Ingram	555-861-2025		
456	Jane	Wright	555-403-1659	555-776-4100	555-403-1659
789	Maria	Fernandez	555-808-9633		

1NF

The designer might, alternatively, retain the single Telephone Number column but alter its domain, making it a string of sufficient length to accommodate multiple telephone numbers:

Customer

Customer ID	First Name	Surname	Telephone Numbers
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659, 555-776-4100
789	Maria	Fernandez	555-808-9633

1NF conditions

- There's no top-to-bottom ordering to the rows.
- There's no left-to-right ordering to the columns.
- There are no duplicate rows.
- Every row-and-column intersection contains exactly one value from the applicable domain (and nothing else).
- All columns are regular [i.e. rows have no hidden components such as row IDs, object IDs, or hidden timestamps].

1NF

A design that is unambiguously in 1NF makes use of two tables: a Customer Name table and a Customer Telephone Number table.

Customer Name

Customer ID	**First Name**	**Surname**
123	Robert	Ingram
456	Jane	Wright
789	Maria	Fernandez

Customer Telephone Number

Customer ID	**Telephone Number**
123	555-861-2025
456	555-403-1659
456	555-776-4100
789	555-808-9633

Prime attribute

A prime attribute, conversely, is an attribute that does occur in some candidate key.

Non-prime attribute

A non-prime attribute is an attribute that does not occur in any candidate key. Employee Address would be a non-prime attribute in the "Employees' Skills" table.

Superkey

A superkey is a combination of attributes that can be used to uniquely identify a database record. A table might have many superkeys.

Candidate key

A candidate key is a special subset of superkeys that do not have any extraneous information in them: it is a minimal superkey.

Examples: Imagine a table with the fields <Name>, <Age>, <SSN> and <Phone Extension>. This table has many possible superkeys. Three of these are <SSN>, <Phone Extension, Name> and <SSN, Name>. Of those listed, only <SSN> is a candidate key, as the others contain information not necessary to uniquely identify records ('SSN' here refers to Social Security Number, which is unique to each person).

2NF

A table that is in first normal form (1NF) must meet additional criteria if it is to qualify for second normal form. Specifically: **a table is in 2NF if and only if, it is in 1NF and no non prime attribute is dependent on any proper subset of any candidate key of the table.** A non prime attribute of a table is an attribute that is not a part of any candidate key of the table.

Functional dependency

In a given table, an attribute Y is said to have a functional dependency on a set of attributes X (written X → Y) if and only if each X value is associated with precisely one Y value. For example, in an "Employee" table that includes the attributes "Employee ID" and "Employee Date of Birth", the functional dependency {Employee ID} → {Employee Date of Birth} would hold. It follows from the previous two sentences that each {Employee ID} is associated with precisely one {Employee Date of Birth}.

2NF

Employees' Skills

Employee	Skill	Current Work Location
Jones	Typing	114 Main Street
Jones	Shorthand	114 Main Street
Jones	Whittling	114 Main Street
Bravo	Light Cleaning	73 Industrial Way
Ellis	Alchemy	73 Industrial Way
Ellis	Flying	73 Industrial Way
Harrison	Light Cleaning	73 Industrial Way

2NF

Employees

Employee	Current Work Location
Jones	114 Main Street
Bravo	73 Industrial Way
Ellis	73 Industrial Way
Harrison	73 Industrial Way

Employees' Skills

Employee	Skill
Jones	Typing
Jones	Shorthand
Jones	Whittling
Bravo	Light Cleaning
Ellis	Alchemy
Ellis	Flying
Harrison	Light Cleaning

2НФ

A B C D E F G

Н_СОТР	ФАМ	Н_ОТД	ТЕЛ	*Н_ПРО*	ПРОЕКТ	Н_ЗАДАН
1	Иванов	1	11-22-33		Космос	1
1	Иванов	1	11-22-33	2	Климат	1
2	Петров	1	11-22-33	*1*	Космос	2
3	Сидоров	2	33-22-11	*1*	Космос	3
3	Сидоров	2	33-22-11	2	Климат	2

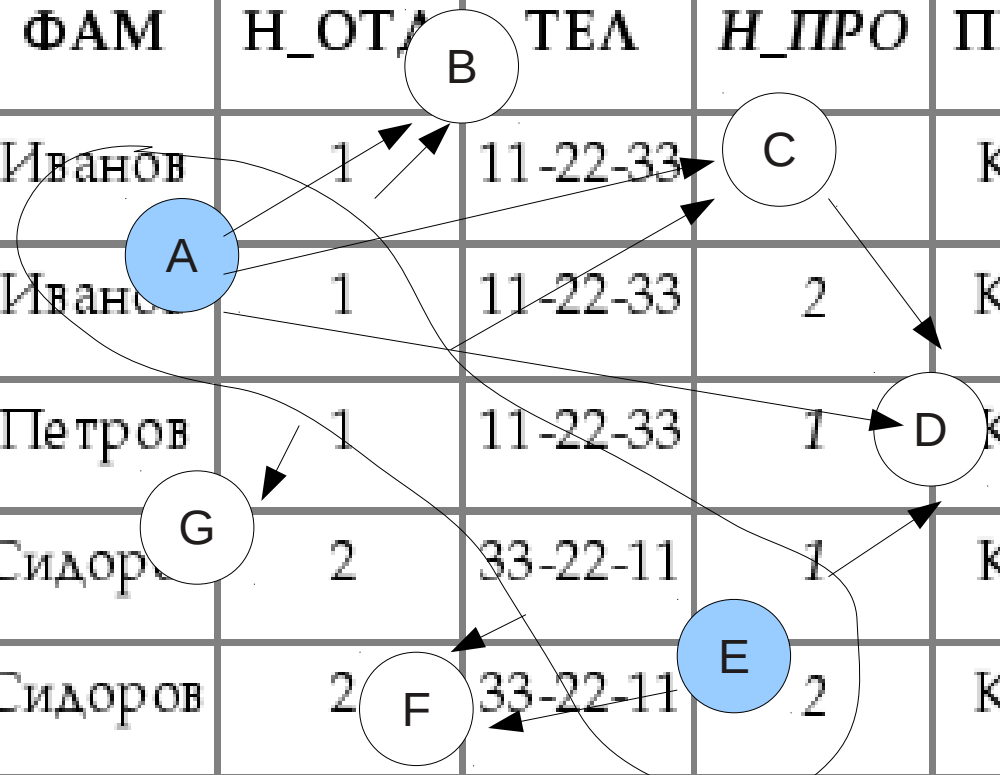


Таблица 1 Отношение СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ

2НФ

Н_СОТР	ФАМ	Н_ОТД	ТЕЛ
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Таблица 2 Отношение СОТРУДНИКИ_ОТДЕЛЫ

Н_ПРО	ПРОЕКТ
1	Космос
2	Климат

Таблица 3 Отношение ПРОЕКТЫ

Н_СОТР	*Н_ПРО*	Н_ЗАДАН
1	*1*	1
1	*2*	1
2	*1*	2
3	*1*	3
3	*2*	2

Таблица 4 Отношения ЗАДАНИЯ

3NF

A table is in 3NF if and only if both of the following conditions hold:

- The relation R (table) is in second normal form (2NF);
- Every non-prime attribute of R is non-transitively dependent (i.e. directly dependent) on every super key of R.

3NF

Orders			
OrderId (PK)	OrderDate	CustomerName	CustomerCity
1	2009-01-01	John Smith	Chicago

BCNF

Boyce–Codd normal form (or BCNF or 3.5NF) is a normal form used in database normalization. It is a slightly stronger version of the third normal form (3NF). BCNF was developed in 1974 by Raymond F. Boyce and Edgar F. Codd to address certain types of anomaly not dealt with by 3NF as originally defined.

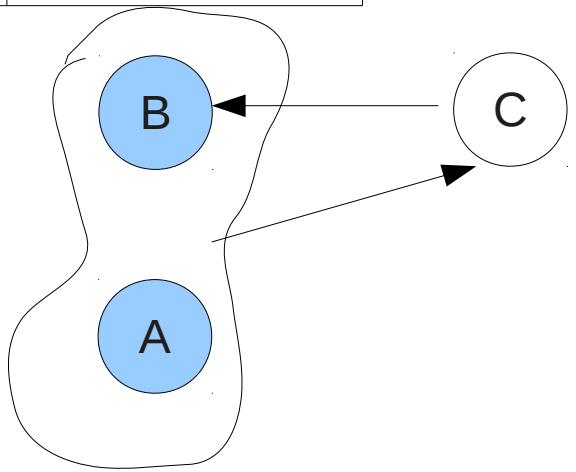
BCNF

A relational schema R is in Boyce–Codd normal form if and only if for every one of its dependencies $X \rightarrow Y$, the following conditions hold:

- $X \rightarrow Y$ is a non trivial functional dependency ($Y \not\subseteq X$)
- X is a superkey for schema R

BCNF

A	B	C
Student	**Discipline**	**Teacher**
100	Algebra	Ivanov
150	Chemistry	Petrov
200	Algebra	Sidorov
250	Algebra	Ivanov
300	Chemistry	Petrov



BCNF

A	C
Student	**Teacher**
100	Ivanov
150	Petrov
200	Sidorov
250	Ivanov
300	Petrov

B	C
Discipline	**Teacher**
Algebra	Ivanov
Chemistry	Petrov
Algebra	Sidorov

4NF

4NF is concerned with a more general type of dependency known as a multivalued dependency. A Table is in 4NF if and only if, for every one of its non-trivial multivalued dependencies X ->> Y, X is a superkey—that is, X is either a candidate key or a superset thereof.

4NF

Pizza Delivery Permutations

Restaurant	Pizza Variety	Delivery Area
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City
A1 Pizza	Stuffed Crust	Springfield
A1 Pizza	Stuffed Crust	Shelbyville
A1 Pizza	Stuffed Crust	Capital City
Elite Pizza	Thin Crust	Capital City
Elite Pizza	Stuffed Crust	Capital City
Vincenzo's Pizza	Thick Crust	Springfield
Vincenzo's Pizza	Thick Crust	Shelbyville
Vincenzo's Pizza	Thin Crust	Springfield
Vincenzo's Pizza	Thin Crust	Shelbyville

4NF

Varieties By Restaurant

Restaurant	Pizza Variety
A1 Pizza	Thick Crust
A1 Pizza	Stuffed Crust
Elite Pizza	Thin Crust
Elite Pizza	Stuffed Crust
Vincenzo's Pizza	Thick Crust
Vincenzo's Pizza	Thin Crust

Delivery Areas By Restaurant

Restaurant	Delivery Area
A1 Pizza	Springfield
A1 Pizza	Shelbyville
A1 Pizza	Capital City
Elite Pizza	Capital City
Vincenzo's Pizza	Springfield
Vincenzo's Pizza	Shelbyville

5NF

Fifth normal form (5NF), also known as Project-join normal form (PJ/NF) is a level of database normalization designed to reduce redundancy in relational databases recording multi-valued facts by isolating semantically related multiple relationships. A table is said to be in the 5NF if and only if every join dependency in it is implied by the candidate keys.

A join dependency *{A, B, … Z} on R is implied by the candidate key(s) of R if and only if each of A, B, …, Z is a superkey for R.

5НФ

X	Y	Z
1	1	2
1	2	1
2	1	1
1	1	1

X	Y
1	1
1	2
2	1

X	Z
1	2
1	1
2	1

Y	Z
1	2
2	1
1	1

6NF (2002, What for?)

A relvar R [table] is in sixth normal form (abbreviated 6NF) if and only if it satisfies no nontrivial join dependencies at all — where, as before, a join dependency is trivial if and only if at least one of the projections (possibly U_projections) involved is taken over the set of all attributes of the relvar [table] concerned.[Date et al.]

Domain/key NF

Domain/key normal form (DKNF) is a normal form used in database normalization which requires that the database contains no constraints other than domain constraints and key constraints.

Normalization

Normalization is a process to increase NF of a relation