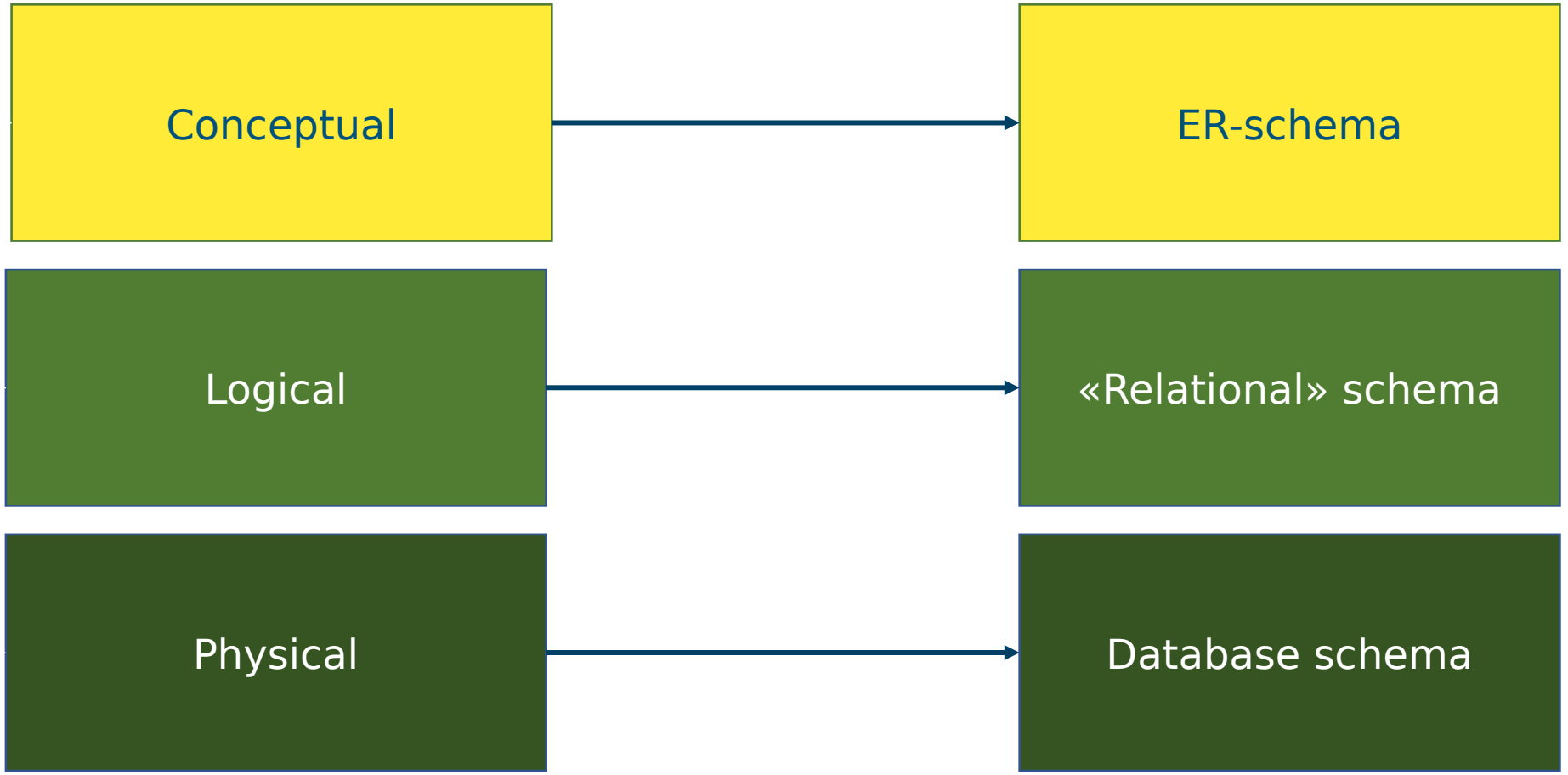
Databases

Lecture 5

Stages of the design



Design stages: characteristics

- Conceptual:
  - Identification of the most general properties of the subject area - entities and relationships between them
- Logical:
  - Improvement of the conceptual scheme
  - Formation of relationships and attributes, primary keys, establishment of functional dependencies
- Physical:
  - Improvement of logical schema
  - Formation of a database schema in relation to a specific DBMS
  - Defining data types, restrictions, creating technical tables (many-to-many, one-to-one, etc.)

ER-schema: one schema in two notations

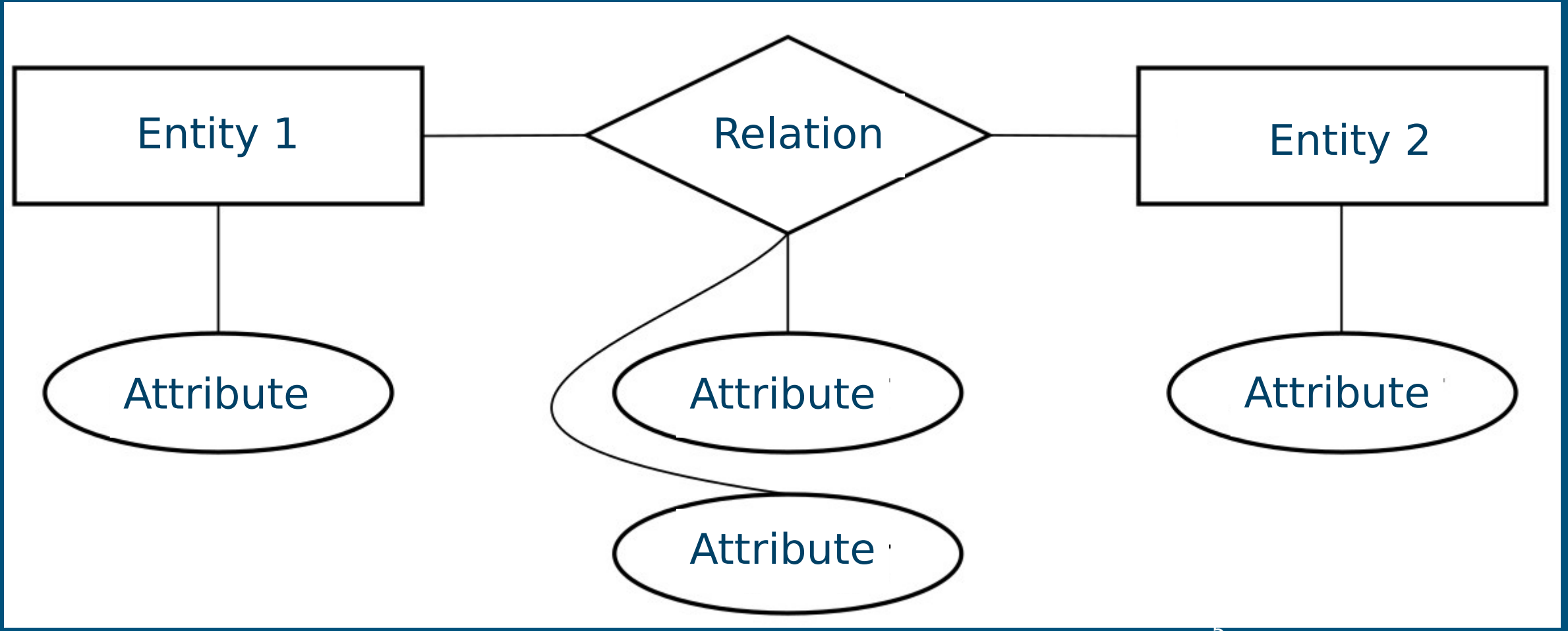
FYI - There are two ER diagram (ER diagram) notations:
Peter Chen's notation:

- Rectangles are entities (expressed as nouns)
- Rhombuses - relationships (verbs)
- Ovals – attributes of entities, properties of relationships (adjectives and adverbs)

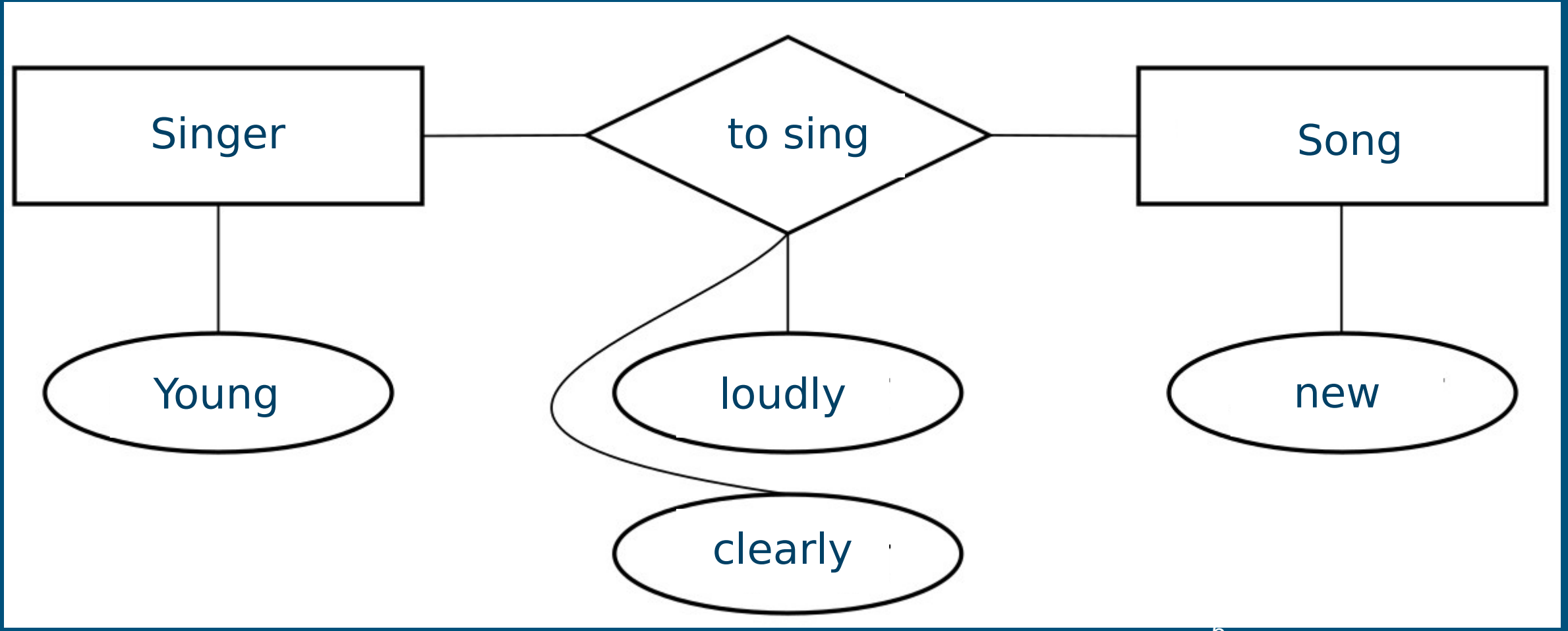
Crow's foot notation:

- Rectangles - entities
- The various arcs between them are relationships. Graphically, the arc expresses the cardinality of the relationship

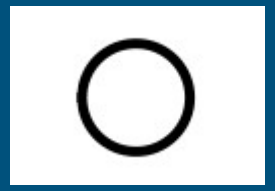
ER-diagram in P.Chen’s notation



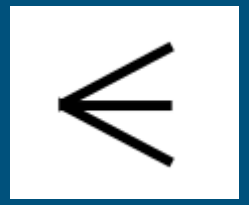
ER-diagram in P.Chen’s notation



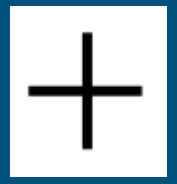
Notation “crow's foot”



Zero



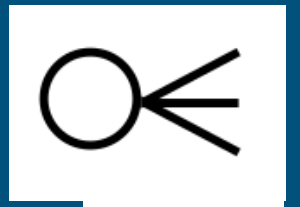
Many



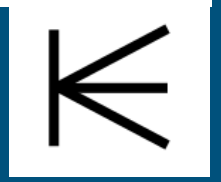
One

Zero or one

One and only one

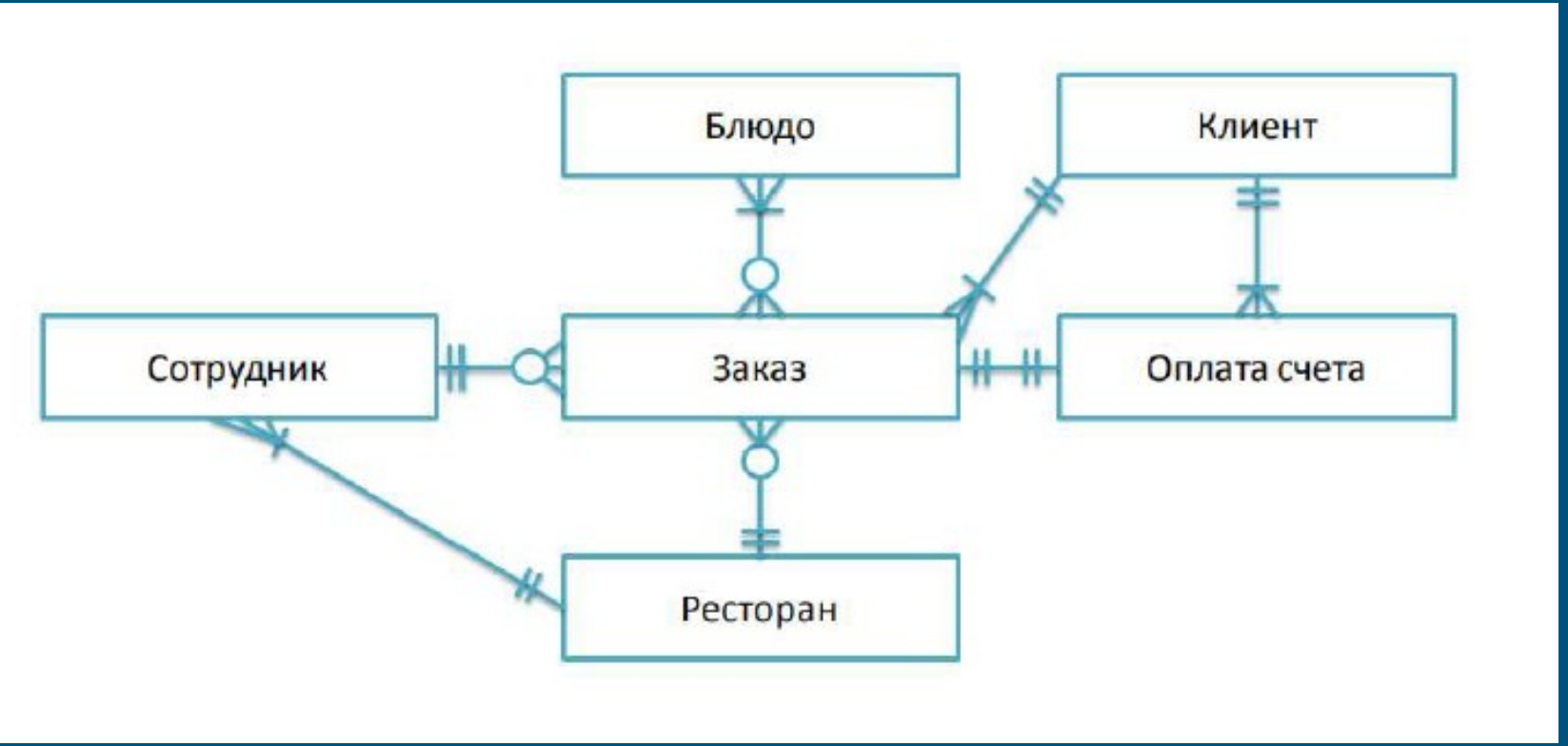


Zero or many

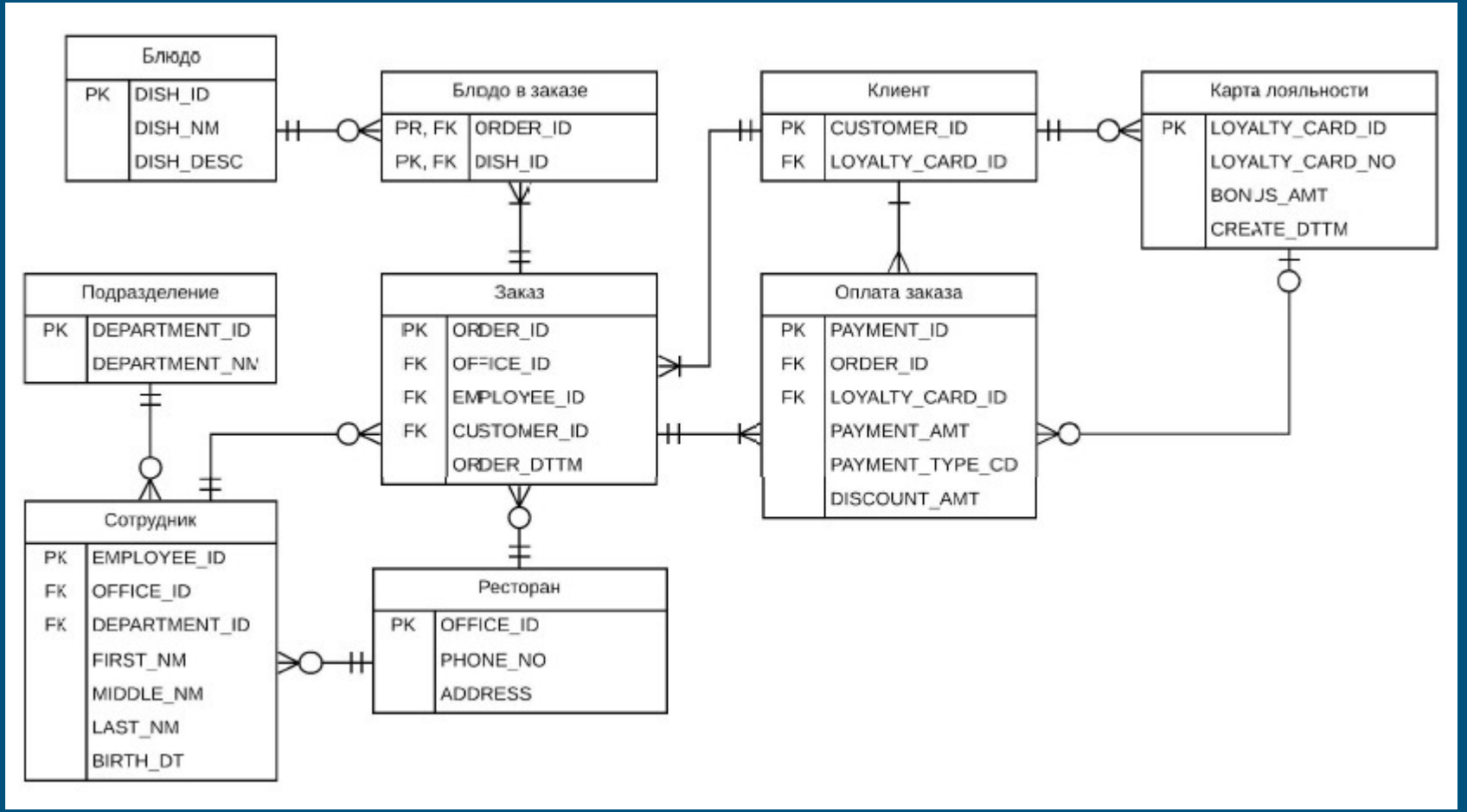


One or many

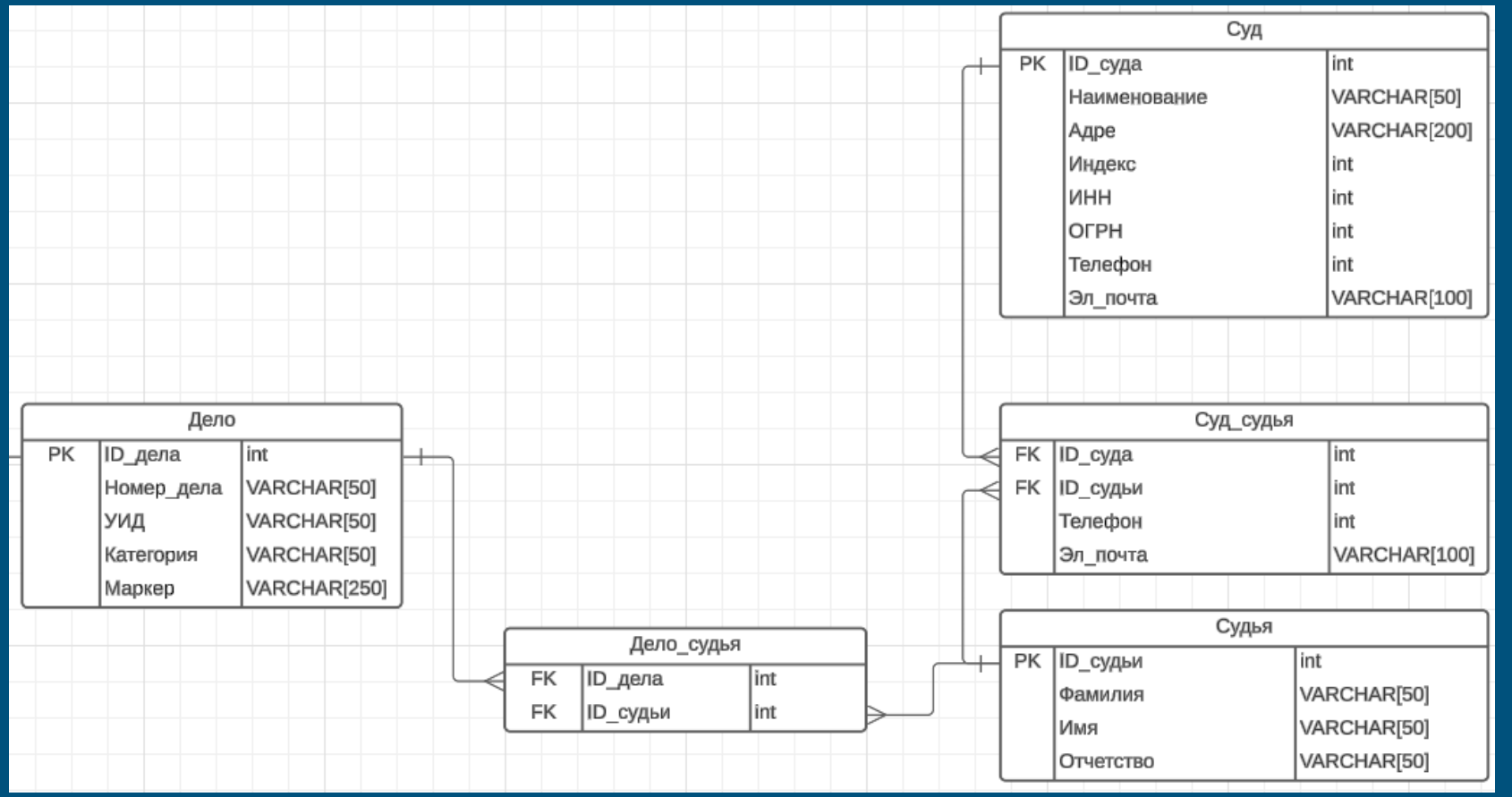
Conceptual model of the restaurant



Logical design: relational schema (logical model of the restaurant)



Physical design: task tracker



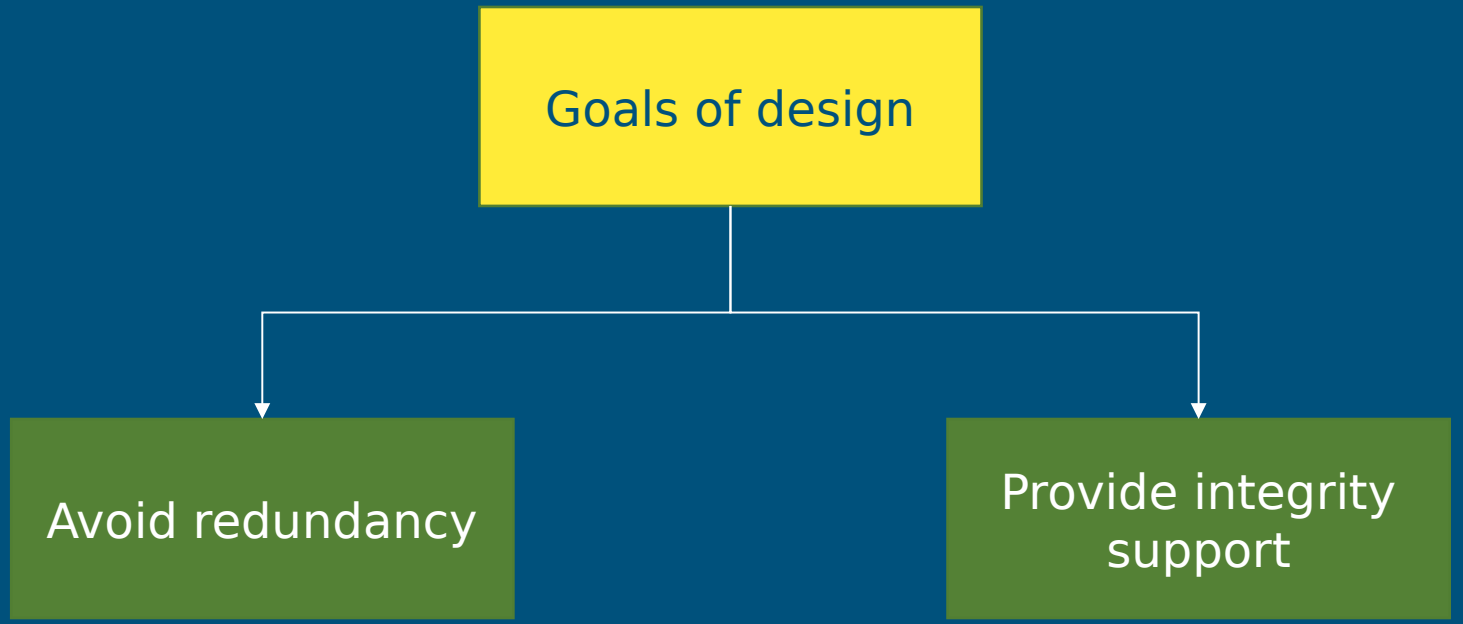
Для чего необходимо проектировать БД?

- Проектирование, помимо очевидных целей определения хранимых данных, также позволяет достичь следующего:
  - Исключить избыточность данных
  - Обеспечить поддержку целостности, и, в частности, ссылочной целостности данных.
- Это важно для поддержания корректности данных при их хранении и обработки
- Если пренебречь качественным проектированием БД, то впоследствии можно столкнуться с различного рода **аномалиями** :
  - добавления,
  - модификации (редактирования),
  - удаления данных
- Для качественного проектирования БД необходимо:
  - Проводить нормализацию данных
  - Использовать средства поддержания (ссылочной) целостности (первичные и внешние ключи, ограничения, проверки)

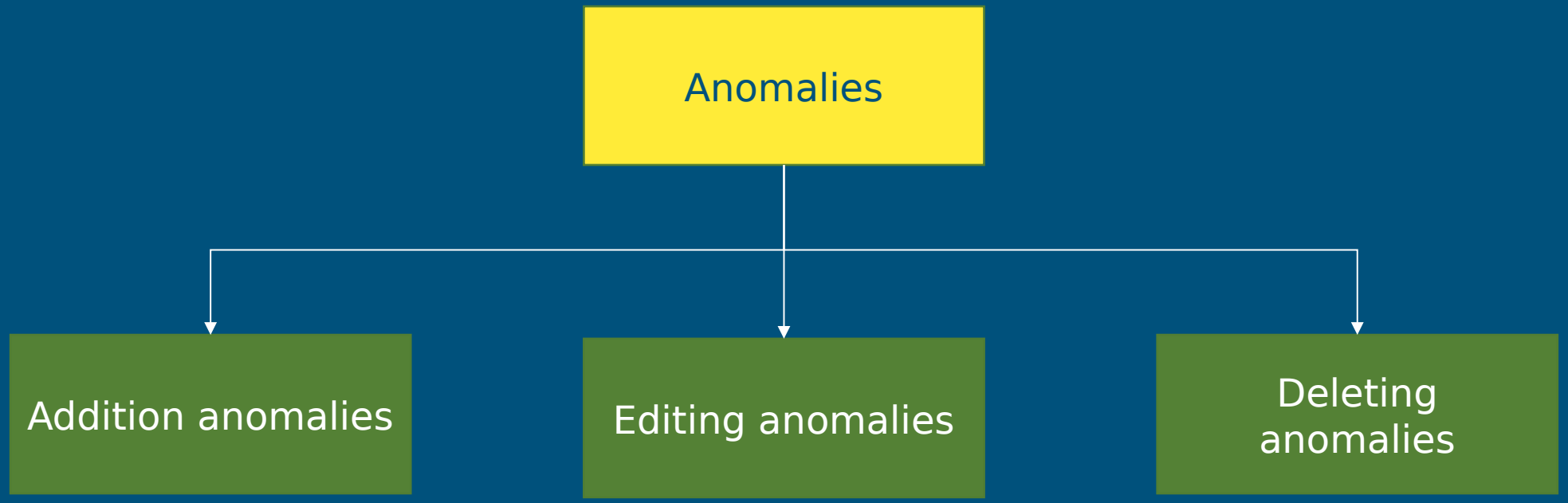
Необходимые в дальнейшем понятия

- **Целостность (сущностей)** – любое отношение должно обладать потенциальным ключом. Поддержание целостности сущностей средствами СУБД: (1) при добавлении записей в таблицу проверяется уникальность их первичных ключей; (2) не допускается изменение значений атрибутов, входящих в первичный ключ
- **Ссылочная целостность** – это необходимое качество реляционной базы данных, заключающееся в отсутствии в любом её отношении внешних ключей, ссылающихся на несуществующие кортежи (по Дж Дейту: база данных не должна содержать каких-либо несогласованных значений внешнего ключа). База данных обладает свойством ссылочной целостности, когда для любой пары связанных внешним ключом отношений в ней условие ссылочной целостности выполняется
- **Избыточность данных** – дублирование данных в базе
- **Аномалия** – ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД. Причиной аномалии является (излишнее) дублирование данных в таблице

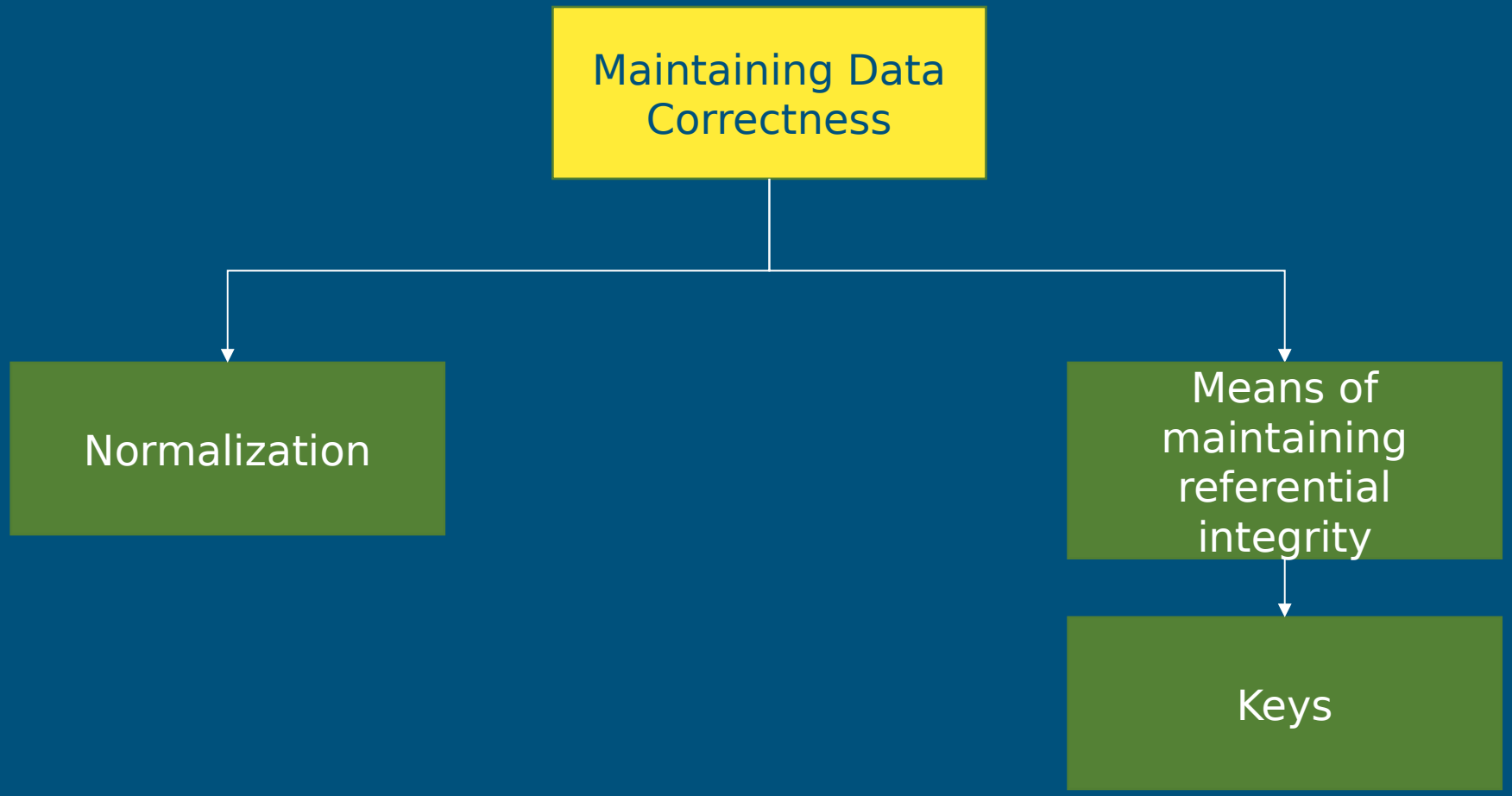
Goals of design



Consequences of low-quality design



Ensuring the data consistency during the design



ANOMALIES OF ADDITION

We cannot add a new entry if the values of the primary keys are unknown

Delivery Number (RC)	Title Product (RK)	Product Price	Quantity	Date deliveries	Title Supplier	Address Supplier
1	Pencil 15		10000	12.10.2017	Supplier_1	Address_1
2	Glue	30	1500	03.03.2018	Purveyor_1	Address_1
2	Notebook	5	10000	03.03.2018	Purveyor_2	Address_2
3	Pen	5	13000	05.03.2018	Purveyor_1	Address_1
3	Notepad	50	20000	05.03.2018	Purveyor_1	Address_1
3	Album	100	25000	05.03.2018	Purveyor_2	Address_2

We have signed a contract with supplier 3:
We cannot add information about him to the table, because there have been no deliveries yet

MODIFICATION ANOMALIES

Changing the data of one record entails the need to change the similar data of some more records

Example: to change the distributor 1’s address, we have to change all the rows

Number delivery (RK)	Name of the product (RK)	Product Price	Quantity	Delivery date	Name of the supplier	Supplier's address
1	Pencil 15		10000	12.10.2017	Supplier_1	Address_1
2	Glue	30	1500	03.03.2018	Supplier_1	Address_1
2	Copybook	5	10000	03.03.2018	Supplier_2	Address_2
3	A pen	5	13000	05.03.2018	Supplier_1	Address_1
3	Notepad	50	20000	05.03.2018	Supplier_1	Address_1
3	Album	100	25000	05.03.2018	Supplier_2	Address_2

ANOMALIES OF DELETION

Deleting certain records carries the loss of information that you did not want to de

Delivery Number (RC)	Product name (RK)	Product price	Quantity	Date deliveries	Name of the supplier	Supplier's address
1	Pencil 15		10000	12.10.2017	Purveyor_1	Address_1
2	Glue	30	1500	03.03.2018	Purveyor_1	Address_1
2	Notebook	5	10000	03.03.2018	Purveyor_2	Address_2
3	Pen	5	13000	05.03.2018	Purveyor_1	Address_1
3	Notepad	50	20000	05.03.2018	Purveyor_1	Address_1
3	Album	100	25000	05.03.2018	Supplier_2	Address_2

We want to delete records of deliveries from supplier 2:
We lose all information about supplier 2, including its address

Referential integrity tools - overview

- Some SQL commands and statements that allow us to maintain (referential) integrity we have already looked at when getting acquainted with DDL - these are the restrictions specified when creating tables with the CREATE TABLE command:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY / FOREIGN KEY
  - CHECK
  - DEFAULT

Referential integrity tools - overview

- In addition to the listed restrictions, you can explicitly set a certain behavior of the DBMS when deleting a certain row or changing a certain value in an external table. External table is a table referenced by a foreign key attribute.
- If a data change causes a row to change but the target columns are not affected by the change, no action is taken.

Referential integrity tools - overview

- In addition to the listed restrictions, you can explicitly set a certain behavior of the DBMS when deleting a certain row or changing a certain value in an external table. External table is a table referenced by a foreign key attribute.
- If a data change causes a row to change but the target columns are not affected by the change, no action is taken.

Referential integrity tools - overview

NO ACTION

- Throw an error indicating that deleting or changing a record will violate a foreign key constraint

RESTRICT

- Throw an error indicating that deleting or changing a record will violate a foreign key constraint

CASCADE

- Delete all rows referencing the deleted record, or change the values in the referencing columns to new values in external columns, in accordance with the operation

Referential integrity tools - overview

SET NULL [(column_name [, ...])]

- Set all referencing columns or a specified subset of referencing columns to null. A subset of columns can only be specified for ON DELETE actions

SET DEFAULT [(column_name [, ...])]

- Set all referencing columns or a specified subset of referencing columns to their default values. A subset of columns can only be specified for ON DELETE actions. (If the default values are non-NULL, the external table must have a row corresponding to the set of these values; otherwise the operation will fail.)

Syntax example

```
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
    customer_id INT REFERENCES customers(customer_id) ON
DELETE CASCADE,
    order_date DATE,
    total_amount DECIMAL
);
```

Referential integrity tools - keys

According to J. Date:

- The definition of a candidate key is essentially just shorthand for some constraint on a relation
- Potential keys are important from a practical point of view: in particular, in the relational model they provide the basic addressing mechanism at the tuple level.
- This means that the only system-guaranteed way to accurately identify a particular tuple is to use a specific candidate key value.

Keys: potential key

A potential key is a subset of relation attributes in a relational data model that satisfies the following requirements:

• uniqueness - there are not and cannot be two tuples of a given relation in which the values of this subset of attributes coincide (are equal)

• irreducibility (minimality) - the potential key does not contain a smaller subset of attributes that satisfies the uniqueness condition. In other words, if any attribute is removed from a potential key, it will lose its uniqueness property

A candidate key always exists, even if it includes all the attributes of the relation (follows from the properties of the relations). There can be several potential keys.

Keys: primary key

Primary key (PRIMARY KEY) is such a potential key of a relationship that is selected as the “primary”

- Any potential key is suitable as a primary key

Alternate key(s) – all other potential keys of the relationship

Keys: foreign key

To reflect the functional dependencies between tuples of different relations, duplication of the primary key of one relation (parent) to another (child) is used. Attributes that are copies of parent relationship keys are called foreign keys.

A foreign key in relation R2 is a non-empty subset FK of the set of attributes of this relation, such that:

1) There is a relation R1 with a potential key PK;

2) Each value of the foreign key FK in the current value of the relation R2 necessarily coincides with the value of the key PK of some tuple in the current value of the relation R1.

The relationships R1 and R2 are not necessarily different.

Keys property -> functional dependency

- Keys allow you to form functional dependencies between relationships
- A functional dependency is a many-to-one relationship between two sets of attributes of a given relation variable

Keys property -> functional dependency

Definition of functional dependence (according to J. Date):

- Let r be a relation, and X and Y be arbitrary subsets of the set of attributes of relation r. Then Y is functionally dependent on X, which is written symbolically as X → Y (read either "X functionally determines Y" or "X is an arrow of Y")
- if and only if every value of the set X of the relation r is associated with exactly one value of the set Y of the relation r.

Keys property -> functional dependency

Definition of functional dependence (according to J. Date):

- In other words, if two tuples of a relation r coincide in the value X, they also coincide in the value Y (⇔ there are no two different tuples that have the same values in the X attribute and different values in the Y attribute).
- The left part of the functional dependence record is called the determinant, the right part is the dependent part.

Examples in one table

Город	Поставщик	Продукт	Количество
Москва	ООО "Ромашка"	Конфеты	100
Санкт-Петербург	ООО "Василек"	Шоколад	300
Казань	ООО "Ландыш	Мороженое	400
Москва	ООО "Ромашка"	Конфеты	200
Санкт-Петербург	ООО "Подорожник"	Шоколад	300

Examples in one table

X -> Y

{City} -> {Product}

{Distributor} -> {City}

{Distributor} -> {Product}

{Amount} -> {Product}

{Amount} -> {City}

{City, Distributor} -> {Product}

{Distributor, Product} -> {City}

Distributor	City	Product	Amount
ООО "Ромашка"	Москва	Конфеты	1000
ООО "Василек"	Санкт-Петербург	Шоколад	3000
ООО "Ландыш	Казань	Мороженое	4000
ООО "Ромашка"	Москва	Конфеты	2000
ООО "Подорожни	Краснодар	Шоколад	3000
ООО "Подорожни	Краснодар	Шоколад	3000

Relation between keys and functional dependencies

Distributor	City	Product
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни	Краснодар	Шоколад
ООО "Подорожни	Краснодар	Шоколад

Relation between keys and functional dependencies

X -> Y

{Distributor} -> {City}

{City} -> {Product}

Information is excessive:

we have duplicates

Distributor	City	Product
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни	Краснодар	Шоколад
ООО "Подорожни	Краснодар	Шоколад

Relation between keys and functional dependencies

How can we get rid of dupes?

Answer: Heath’s theorem

Distributor	City	Product
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни	Краснодар	Шоколад
ООО "Подорожни	Краснодар	Шоколад

Relation between keys and functional dependencies

Heath’s theorem.

Let’s consider a relation R(A,B,C), where A,B,C are attributes. If R satisfies the functional dependency A -> B, then R is a union of R’s projections on attributes (A,B) and (A,C)

Relation between keys and functional dependencies

A = {Distributor}

B = {City}

C = {Product}

-> {Distributor, City} , {Distributor, Product}}

Distributor	City	Product
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни	Краснодар	Шоколад
ООО "Подорожни	Краснодар	Шоколад

Relation between keys and functional dependencies

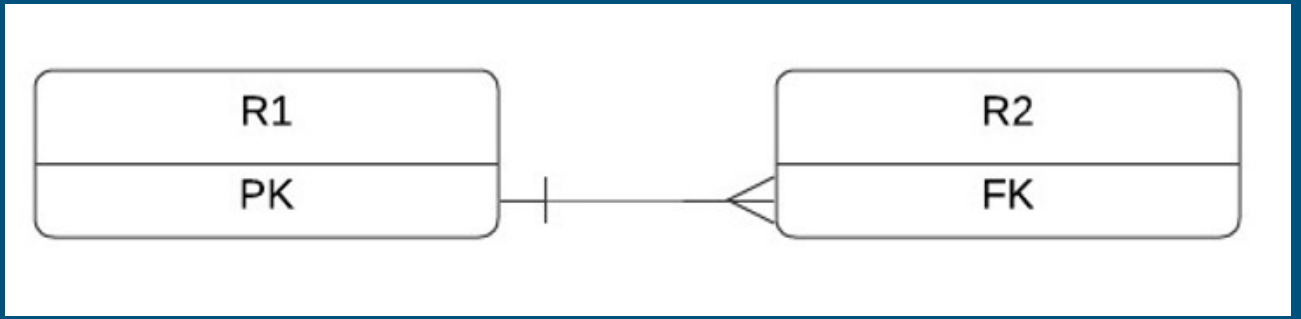
Distributo	City	Produc
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни	Краснодар	Шоколад
ООО "Подорожни	Краснодар	Шоколад

Distributor	City
ООО "Ромашка"	Москва
ООО "Василек"	Санкт-Петербург
ООО "Ландыш	Казань
ООО "Подорожни	Краснодар

Distributor	Product
ООО "Ромашка"	Конфеты
ООО "Василек"	Шоколад
ООО "Ландыш	Мороженое
ООО "Подорожни	Шоколад

Relation between keys and functional dependencies

- In this case, the {Distributor} attribute can be selected as the primary key in the {Distributor, City} relation, and in the {Distributor, Product} relation, this attribute can have a foreign key constraint.
- Generalizing the concept of functional dependence, we can (loosely) say that a primary key defines the functional dependence of the table containing it in relation to the table containing the foreign key. That is, if the values of “attributes – foreign keys” are equal, they are always equal to the value of the “attribute – primary key”.



NORMAL FORM

- Normal form is a property of a relationship in a relational data model that characterizes it from the point of view of redundancy, potentially leading to logically erroneous results of sampling or data modification
- The normal form is defined as a set of requirements that the relation must satisfy
- Bringing the database to normal form – normalization
- Each following form includes the limitations of the previous ones

DATABASE NORMALIZATION

- Exclusion of certain types of redundancy
- Elimination of some anomalies* Updates
- Development of a database project that is:
  - High-quality representation of the real world
  - Intuitive
  - Easy to expand in the future
- Simplification of the procedure for applying the necessary integrity constraints

DATABASE NORMALIZATION

Intended:

- Minimizing logical redundancy
- Reducing potential inconsistency

Not intended for:

- Decrease/increase in DB performance
- Decrease / increase in the physical volume of the database

Что такое нормализация и для чего она нужна

- Нормализация БД производится за счет декомпозиции отношения (см. выше теорему Хита).
- Декомпозиция – разложение исходной переменной отношения на несколько эквивалентных. Декомпозиция обратна соединению
- **Декомпозиция называется декомпозицией без потерь или правильной , если она обратима**

NORMAL FORMS

- **First Normal Form (1NF)**
- **Second Normal Form (2NF)**
- **Third Normal Form (3NF)**
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form / Projection-junction Normal Form (5NF/PJNF)
- Domain-key Normal Form (DKNF)
- Sixth normal Form (6NF)

1NF

- Переменная отношения находится **в первой нормальной форме** тогда и только тогда, когда значения всех атрибутов отношения атомарны. То есть отношение находится в 1 NF , если все его атрибуты являются простыми. Все используемые домены содержат только скалярные значения.

- Неформальное определение:
  - Все строки должны быть различными
  - Все элементы внутри ячеек должны быть атомарными, т.е. не должны быть списками. Другими словами, элемент является атомарным, если его нельзя разделить на части, которые могут использовать в таблице независимо друг от друга

- Любое реляционное отношение находится в 1НФ на основании свойств реляционного отношения.

THE FIRST NORMAL FORM

- A relation variable is in the first normal form if and only if the values of all attributes of the relation are atomic.
- A relation is in 1NF if all its attributes are simple. All domains used contain only scalar values.

1NF

Город	Улица
Москва	Моховая, Воздвиженка, Неглинная
Санкт-Пет	Невский проспект, Миллионная, Рубинштейна
Казань	Кремлевская, Баумана, Пушкина

1NF

Город	Улица
Москва	Моховая, Воздвиженка, Неглинная
Санкт-Петербург	Невский проспект, Миллионная, Рубинштейна
Казань	Кремлевская, Баумана, Пушкина

Город	Улица
Москва	Моховая
Москва	Воздвиженка
Москва	Неглинная
Санкт-Петербург	Невский проспект
Санкт-Петербург	Миллионная
Санкт-Петербург	Рубинштейна
Казань	Кремлевская
Казань	Баумана
Казань	Пушкина

THE SECOND NORMAL FORM

- A relation variable is in the second normal form if and only if it is in the first normal form, and each non-key attribute is minimally functionally dependent on a potential key
- The functional relationship between the sets of attributes X and Y means that for any valid set of tuples , the following is true in this respect: if two tuples coincide in the value of X, then they coincide in the value of Y
- Minimal functional dependency means that the primary key does not contain a smaller subset of attributes, relations which can also be derived from this functional dependency

2NF

An informal definition for a table with a single key:

- The table must be in first normal form
- Any of its fields that are not part of the primary key functionally depends only on all the attributes of the key and does not depend on its individual attributes

If a table with a single key is in first normal form and has a unique id for each row, then it is in second normal form.

2NF

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

2NF

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

2NF

Название группы	Название CD-диска	Название песни
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Название группы	Название песни	Автор слов	Композитор
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Klaus Meine

2NF – объяснение применительно к теореме Хита

- Нам нужны три группы атрибутов и ФЗ между двумя из них:
- Если A → B, то R = {AB} ⋈ {AC}, следовательно если
  - ФЗ: {Название группы, Название песни} → {Автор слов, Композитор}
- То:
  - A – {Название группы, Название песни}
  - B – {Автор слов, Композитор}
  - C – {Название CD-диска}
- Тогда по теореме Хита:
  - Если A → B, то R = {AB} ⋈ {AC} или

  R = {Название группы, Название песни, Автор слов, Композитор} ⋈
  ⋈ {Название группы, Название песни, Название CD-диска}