Databases

Lecture 1
Introduction

About the course

- about 15 lectures and practical lessons
- 3 tests (blocking)
- 5 homeworks
- quizzes
- project (blocking)
- differential mark

Grade:

0.1 * (sum of quizzes) + 0.2 * (sum of tests) + 0.2 * (sum of HW) + 0.5 * (project) + 0.2 * (final test) + 0.1*(bonus)

Database

- a set of data stored in accordance with the data schema, which is manipulated in accordance with the rules of data modeling tools
- a collection of data organized according to a conceptual structure describing the characteristics of this data and the relationships between them, and such a collection of data that supports one or more application areas

Why we need databases

storage and processing of a large amount of information

data sharing

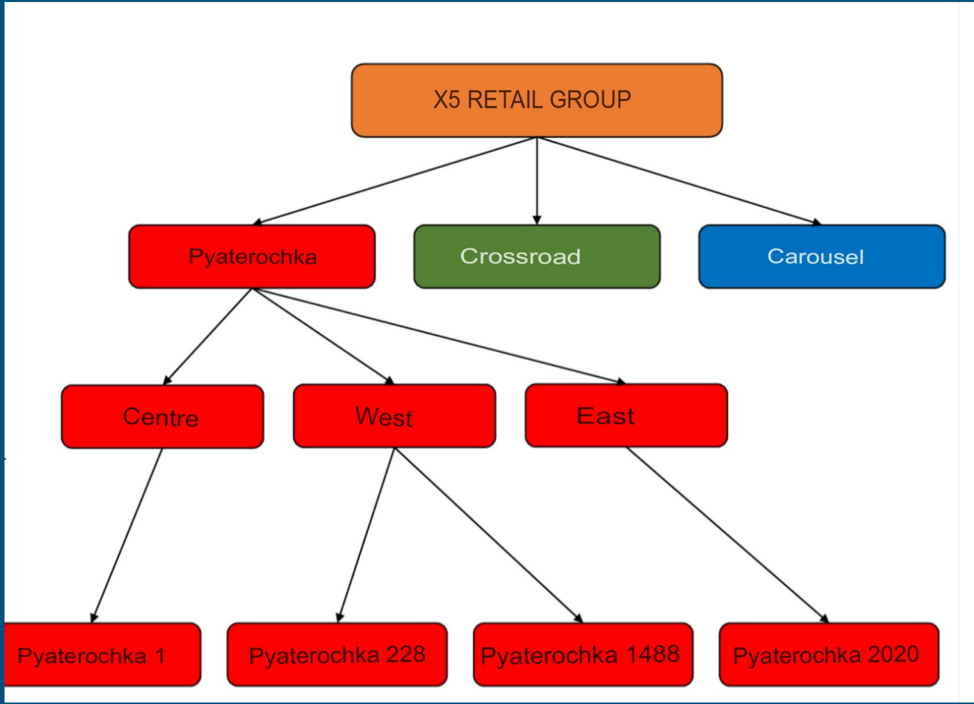
Data Models

- ★ Hierarchical data model
- ★ Network Data Model
- ★ Relational data model

Hierarchical data model

IBM, 1960s

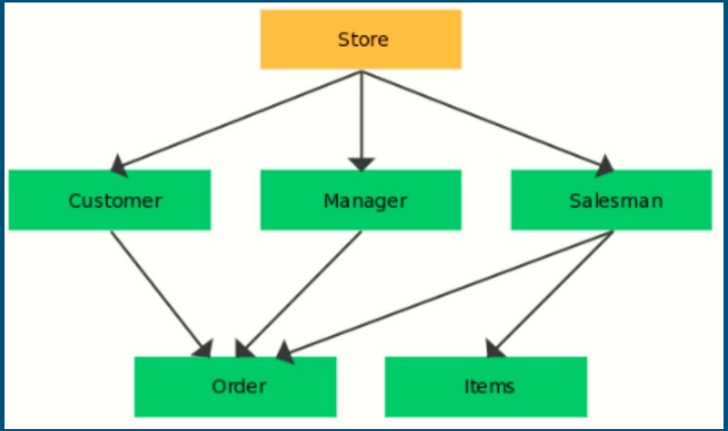
- Tree structure of records
- A descendant has exactly 1 ancestor
- Descendants of a common ancestor – twins

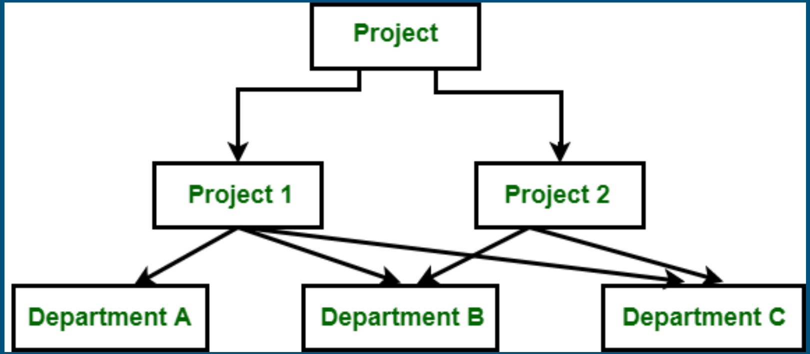


Network data model

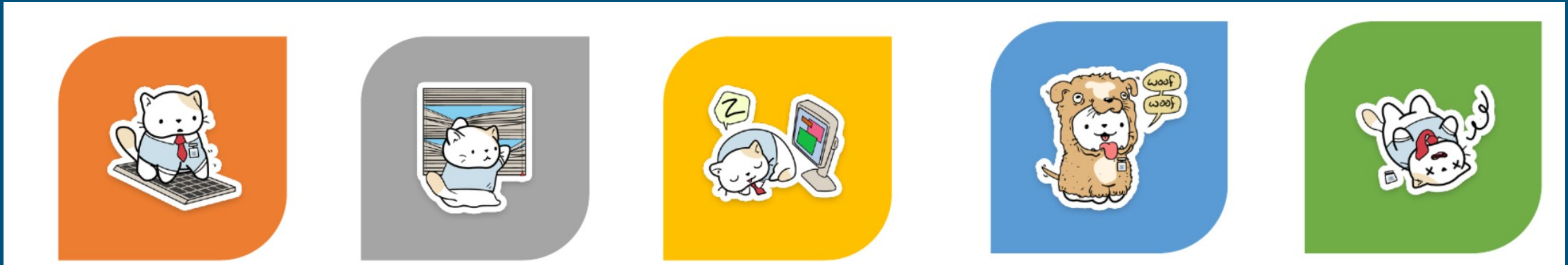
Charles Bachman, 1969

- The structure of records in the form of a graph
- Extends the hierarchical data model
- A descendant can have more than 1 ancestor





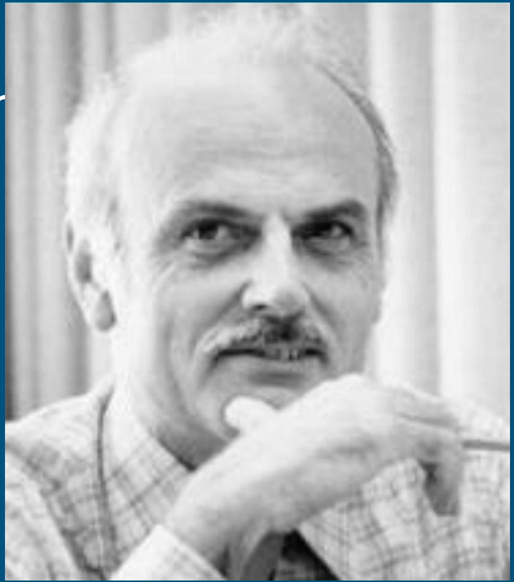
Disadvantages



- High complexity and rigidity of DB schema
- Lack of flexibility
- Performing even simple queries is a complex process
- Dependence on physical data organizations
- Low efficiency

Relational data model

- 1969-1970 E. Codd
- June 1970 "A Relational Model of Data for
- Large Shared Data Banks"
- The model is based on mathematics and
- Logical data model
- Does not depend on physical structures



Basic concepts

- A **domain** is a set of valid values.
- The **attribute** is the name of the domain.
- A **tuple** is an ordered set of fixed length.
- The **Cartesian product** (of sets A = (a1, a2,...) and B = (b1, b2,...)) is the set of pairs: A B = {(a,b): a ∈ A & b ∈ B}.
- The **arity** of a relation is the number of its elements (the number of attributes).

Example: relation

relation

ID	FIRST_NM	LAST_NM	PHONE_NO	CREATE_DT
1	Ivan	Ivanov	79039065432	2017-08-12
101	Sergey	Serov	79612345623	2003-05-14
1006	Piotr	Petrov	79013724683	2018-06-24
70009	Nikolay	Sidorov	79262345401	2013-12-16

Example

header of relation

ID	FIRST_NM	LAST_NM	PHONE_NO	CREATE_DT
1	Ivan	Ivanov	79039065432	2017-08-12
101	Sergey	Serov	79612345623	2003-05-14
1006	Piotr	Petrov	79013724683	2018-06-24
70009	Nikolay	Sidorov	79262345401	2013-12-16

tuple

Example

ID	FIRST_NM	LAST_NM	PHONE_NO	CREATE_DT
1	Ivan	Ivanov	79039065432	2017-08-12
101	Sergey	Serov	79612345623	2003-05-14
1006	Piotr	Petrov	79013724683	2018-06-24
70009	Nikolay	Sidorov	79262345401	2013-12-16

domain: natural numbers

Cartesian product

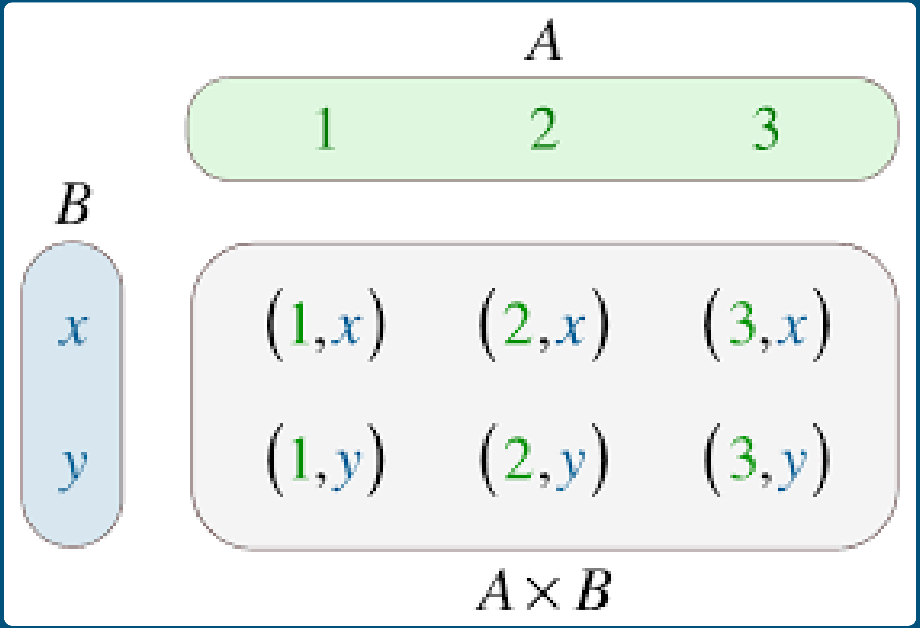
A and B are sets

- $A = (a_1, a_2, \dots)$
- $B = (b_1, b_2, \dots)$

The **Cartesian product** of sets A and B is the set of pairs:

$$A \times B = \{(a, b) : a \in A \,\&\, b \in B\}$$

Cartesian product



Extended cartesian product

We have N sets, where N > 1

$$S_i = (s_{i1}, s_{i2}, \dots, s_{ij}, \dots)$$

An **extended Cartesian product** of N sets is a set of the form:

$$S_1 \times S_2 \times \cdots \times S_N = \{(x_1, x_2, \dots, x_N) : x_i \in S_i, i = \overline{1, N}\}$$

An element of such a set is called a **tuple** $(x_1, x_2, \dots, x_N)$

Extended Cartesian product

Q

Customers

CustomerId	Name
1	Shree
2	Kalpana
3	Basavaraj

Orders

OrderID	CustomerId	OrderDate
100	1	2014-01-29 23:56:57.700
200	4	2014-01-30 23:56:57.700
300	3	2014-01-31 23:56:57.700

R

Q x R = Z

CustomerId	Name	OrderID	CustomerId	OrderDate
1	Shree	100	1	2014-01-30 23:48:32.850
2	Kalpana	100	1	2014-01-30 23:48:32.850
3	Basavaraj	100	1	2014-01-30 23:48:32.850
1	Shree	200	4	2014-01-31 23:48:32.853
2	Kalpana	200	4	2014-01-31 23:48:32.853
3	Basavaraj	200	4	2014-01-31 23:48:32.853
1	Shree	300	3	2014-02-31 23:48:32.853
2	Kalpana	300	3	2014-02-31 23:48:32.853
3	Basavaraj	300	3	2014-02-31 23:48:32.853

Coupling of tuples

x and y are tuples $$x = (x_1, x_2, \dots, x_n)$$ $$y = (y_1, y_2, \dots, y_m)$$

Then the **coupling** of tuples x and y will be a tuple of dimension n + m:

$$x \times y = (x_1, \dots, x_n, y_1, \dots, y_m)$$

Basic concepts

Domains $D_1, D_2, \ldots, D_N$

A list of attributes is given, so that each domain D_i corresponds to an attribute A_i defined on this domain.

Then by the **relation** defined on attributes (domains) is a subset of the extended Cartesian product of these domains:

$$R \subseteq D_1 \times \cdots \times D_N$$

Basic concepts

- The **arity** of a relationship is the number of attributes.
- **Relationship title** – a list of attributes.
- The set of tuples that make up the relationship is the **body** of the relationship.
- A domain is called **composite** if it is an extended Cartesian product of a finite number of simple domains.
- We will say that two simple domains 1 and 2 are **compatible** if they either coincide, or $2 \subseteq 1$ or $2 \subseteq 1$

Relationship Properties

- No two tuples are the same;
- The order of tuples is not defined;
- The order of attributes is not defined.

Example of a relational data model

Relationship. In general, the table is not a relation. Why?

ID	FIRST_NM	LAST_NM	PHONE_NO	CREATE_DT
1	Ivan	Ivanov	79039065432	2017-08-12
101	Sergey	Serov	79612345623	2003-05-14
1006	Piotr	Petrov	79013724683	2018-06-24
70009	Nikolay	Sidorov	79262345401	2013-12-16

Relational algebra

A family $\mathfrak{A} \subset 2^X$ of subsets of a set X (the carrier of an algebra) is called an algebra if it satisfies the following properties:

$\emptyset \in \mathfrak{A}$

1. If $A \in \mathfrak{A}$, then $X \setminus A \in \mathfrak{A}$
2. If $A, B \in \mathfrak{A}$, then $A \cup B \in \mathfrak{A}$.

Relational algebra:
Carrier – a set of (all possible) relations of various (finite) orders

Set-theoretic operations

Applicable to compatible relationships:

- Association
- Difference
- Intersection

Hereafter Cartesian product == extended Cartesian product, unless otherwise specified

Relational operations: limitation

Construction of a new relationship, which includes tuples that satisfy a given condition.

- R – the specified ratio,
- A and B – lists of attribute identifiers,

$\theta \in \{=, \neq, <, >, \geq, \leq\}.$

Problem: it is permissible only to compare the values of (composite) attributes within the same tuple

$$R[A\theta B] = \{r | r \in R \,\&\, (r[A]\theta r[B])\}$$

Relational operations: limitation

$$R[\lambda\theta\mu] = \begin{cases} R[A\theta B], & \text{if } \lambda = A, \mu = B \\ R[\alpha\theta B] = (R \times \_(A_\alpha)\{(\alpha)\})[A_\alpha \theta B], & \text{if } \lambda = \alpha, \mu = B \\ R[A\theta\beta] = \left(R \times \_\left(B_\beta\right)\{(\beta)\}\right)\left[A\theta B_\beta\right], & \text{if } \lambda = A, \mu = \beta \\ R[\alpha\theta\beta] = \left(R \times \_(A_\alpha)\{(\alpha)\} \times \_\left(B_\beta\right)\{(\beta)\}[A_\alpha \theta B_\beta]\right), & \text{if } \lambda = \alpha, \mu = \beta \end{cases}$$

person_name	score_amt
Ivan	10
Piotr	3
Nikolay	15
Sergey	20
Ilia	0
Anna	5
Maxim	30
Dmitry	7

$Q = R[score\_amt \geq 15]$

person_name	score_amt
Ivan	10
Piotr	3
Nikolay	15
Sergey	20
Ilia	0
Anna	5
Maxim	30
Dmitry	7

$Q = R[score\_amt \geq 15]$

person_name	score_amt
Nikolay	15
Sergey	20
Maxim	30

Relational operations: projection

Building a new relationship with a given list of attributes.

$$R[L] = \{r[L] | r \in R\}$$

first_nm	last_nm	score_amt
Ivan	Ivanov	10
Piotr	Petrov	3
Nikolay	Ivanov	15
Sergey	Serov	20
Ilia	Ivanov	0
Anna	Petrova	5
Maxim	Serov	30
Dmitry	Petrov	7

$$Q = R[last\_nm]$$

first_nm	last_nm	score_amt
Ivan	Ivanov	10
Piotr	Petrov	3
Nikolay	Ivanov	15
Sergey	Serov	20
Ilia	Ivanov	0
Anna	Petrova	5
Maxim	Serov	30
Dmitry	Petrov	7

$Q = R[last\_nm]$

last_nm
Ivanov
Petrov
Serov
Petrova

Relational operations: connection

The composition of a Cartesian product of two relations followed by a constraint on a given condition.

- $\theta \in \{=, \neq, <, >, \leq, \geq\}$
- relation $R_1(A_1, \dots, A_n)$
- relation $R_2(B_1, \dots, B_m)$

$$R_1 \left[R_1[A_i] \theta R_2[B_j] \right] R_2 = \{ r_1 \times r_2 | r_1 \in R_1 \;\&\; r_2 \in R_2 \;\&\; r_1[A_i] \theta r_2[B_j] \}$$

As A_i, B_i , you can use attribute lists. Natural connection: removes an extra attribute

R1

id	uid	task_code	comment_txt
1	123456	A	awful
2	101010	A	excellent
3	123456	B	terrible
4	101010	B	outstanding
5	123456	C	so-so
6	101010	C	unbelievable
7	101010	D	the best of the best

R2

uid	student_nm
123456	You
101010	Son of your mother's friend
136789	Bob

Q = R1[R1[UID] = R2[UID]]R2

id	uid	task_code	comment_txt
1	123456	A	awful
2	101010	A	excellent
3	123456	B	terrible
4	101010	B	outstanding
5	123456	C	so-so
6	101010	C	unbelievable
7	101010	D	the best of the best

uid	student_nm
123456	You
101010	Son of your mother's friend
136789	Bob

Q = R1[R1[UID] = R2[UID]]R2

id	uid	task_code	comment_txt	student_nm
1	123456	A	awful	You
2	101010	A	excellent	Son of your mother's friend
3	123456	B	terrible	You
4	101010	B	outstanding	Son of your mother's friend
5	123456	C	so-so	You
6	101010	C	unbelievable	Son of your mother's friend
7	101010	D	the best of the best	Son of your mother's friend

Division

R

id	series_nm	channel_nm
0	The Simpsons	RenTV
0	The Simpsons	2x2
0	The Simpsons	CTC
1	Family Guy	RenTV
1	Family Guy	2x2
2	Duck Tales	CTC
2	Duck Tales	2x2

S

channel_nm
RenTV
2x2

Q = R/S

We want to get the series
from relation 1,
which were broadcast on all channels
from relation 2

id	series_nm
0	The Simpsons
1	Family Guy