

Task 6-9, 11, 15

Laboratory work 1

1.6) Teams, VK, Steam, Telegram, Uninet, Instagram

1.7) File-processing system

DBMS

1. (Data redundancy and inconsistency)

Files may have different structures,
be written in different languages,
be duplicated, copies of data
may no longer agree.

The information
is structured, its
elements have clear
connections to each other
and have the required data types.

It is difficult to access data:
(special programs ^{are needed} for quick extractions
of data with particular conditions)

Data can be easily
accessed and sorted.

Data isolation:

Files may be in different formats,
so it is difficult to write a program
for an extraction of the appropriate data.

All files have the
required data type
and can be easily
accessed.

Integrity problem:

It is difficult to integrate new
constraints into the program

The program can
be easily modified

Atomicity problem:

it is difficult to ensure the restoring of data after the failure.

Concurrent-access anomalies:

the system acts incorrectly if the ^{same} information is updated by several users at the same time.

Security problems:

all the users have access to all the data.

Operations are atomic.

They happen in their entirety or not happen at all.

System provides a supervision to prevent such kind of failures.

System has special constraints for different users. Abstract view.

1.2) Physical data independence means that different changes in the lower level of abstraction won't modify the view of the higher levels. It allows us not to rewrite applications after the minor changes in physical schema (if they don't depend on it), and also allows us to hide these complexities from the higher-level users. So, developers don't have to be concerned with the lower-level details.

1.9)

1. Atomicity: transaction happens in its entirety or doesn't happen at all.

It would be impossible to restore data after a failure without it.

2. Consistency: the information has to be preserved during the transaction.

Without it, the changes in the system may be saved incorrectly.

3. Durability: the information has to be preserved after the transaction.

Without it, all the changes in the system may be lost after a ^{system} failure.

4. Security: all the users have their own abstract view of the database.

Users would see all the data stored in a database without any permission without it.

5. Data manipulation: database allows us to easily extract and manipulate the data.

Database would be useless without such abilities.

1.14) Transaction management : // Storage manager
concurrency-control manager. // file manager
consistency constraints

1.15) ID | mail | name | date of birth | avatar
ID | mail | balance | bonuses | history of purchases
ID | mail | list of friends | subscribers | subscriptions

Relational Model

(Tables)