MINISTRY OF DEVELOPMENT OF INFORMATION TECHNOLOGIES AND COMMUNICATIONS OF THE REPUBLIC OF UZBEKISTAN

TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES NAMED AFTER MUHAMMAD AL-KHWARIZMI NUKUS BRANCH

Department of Information technologies

Computer engineering (Computer engineering) direction

Permitted to defence

Head of the department

Aytmuratov B. I.

_____

«___»_____ 2017 y.

DEVELOPING THE DATABASE "BANK OF PROBLEM" TO ORGANIZE THE THEMES OF GRADUATE QUALIFICATION WORK

GRADUATE QUALIFICATION WORK

Graduate : _____ Seytimbetov M. T.

Supervisor: _____ Bisenbaev J.

Nukus 2017

CONTENTS

INTRODUCTION

Nowadays choosing actual themes of diploma work of students in Higher educational institutions is one of the main tasks of universities. While choosing the topics correlation to the area, actuality and getting productive results from targets are paid great attention. First of all, actual topics should be found out, learnt and fixed. To sort out and gather problems in one stock is handy way for choosing a topic. If the number of the topics increases, it will make some difficulties for users to work with, control and use. To seek for necessary information from the stock of topic problems, to change it, to use the ways of adding new one or deleting is one of the tasks that users faced with. Of course, nowadays using the area of information technologies is giving efficient results for alleviating such kind of problems. To use database at any sphere will increase its efficiency.

The following systems must be done:

- To provide the general reception or explicated reports of the work on total;
- To allow easy way for finding out the most important factor trends of the change;
- To provide the reception of critical information on time, without essential delay;
- To execute an exact and complete analysis of data.

Modern DBMS basically are exhibits Windows, since given ambience allows in greater depth to use the possibility personal COMPUTER, than ambience DOS. The Reduction of the cost large powered PC has conditioned not only broad transition to ambience Windows, where developer of software can in degree less to take care of distribution resource, but has also done software PC as a whole and DBMS in particular more critical to hardware resource COMPUTER.

Amongst the most bright representatives managerial system database possible to note: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, and also database of Microsoft

SQL Server and Oracle, used in exhibits, built on technologies "client-server". Practically, beside any modern DBMS exists the analogue, produced other company, having similar application and possibility, any exhibit capable to work with many format of the presentation data, realize the export and import given due to presence of the large number converter. Generally accepted, also, are a technologists, allowing use the possibility of the other exhibits, for instance, word processors, package of the building graph etc., and built-in versions of the languages high level (more often - an idioms SQL and/or VBA) and facility of the visual programming interface under development exhibits. So already has a no vital importance on what language and on base what package is written concrete exhibit, and what format given in he is used. Moreover, standard "de facto" became "quick development of applications" or RAD (from english Rapid Application Development), founded on broadly proclaimed in literature "open approach" that is to say demand and possibility of the use the different applied programmes and technologies for development is more flexible and powerful systems data processing. So in one row with "classical" DBMS all are more often mentionned programming languages Visual C++ and With#, which allow quickly to create the necessary components of applications critical on velocities of the functioning, which it is difficult, but sometimes impossible develop the facility "classical" DBMS. The Modern approach to control database implies also broad use to technologies "client-server"

Summing up all that has just been said we can conclude that by the results of using database one can get great efficiency. According to this we put forward the aim of working out “Problem base” database.

The aim of this qualification paper is to create database for doing effective work of choosing topics for qualification thesises. The University of Tashkent information technologies named after Al-Khorazmiy Nukus branch was chosen as an object of the qualification paper.

According to this general aim the following particular tasks are put forward:

- To investigate the work of doing qualification paper in High educational institutions;
- To investigate the process of choosing topics for qualification paper;
- To investigate the source of problems for topics;
- To find out the structure of the problem;
- To find out essential information for database;
- To choose the system of leading database;
- To choose the tools for the interface of database;
- To design and work out the database;
- To work out the Interface programme;
- To test the programme;
- To get the results and analyse them.

The materials of the qualification paper are stated in the order how it was done. The present qualification paper consists of an introduction, 3 chapters, a conclusion, apps and a bibliography.

The first chapter studies the model about the sources of problems, its appearance, the person who adds a topic to database and how to use it. And also the requirement of database is investigated.

The second chapter studies the conception of database, its structure and its task. Database will be designed.

In the third chapter necessary instrumental tools will be chosen, database will be worked out, for using it the appearance of the interface will be planned and the programme will be done.

In conclusion there are the results from using the database and the goals gained by doing the qualification paper.

CHAPTER 1. ESSENTIAL INFORMATION AND INFORMATION DATA

1.1 Description of the application sphere

The Development of information technology is accompanied two more curious trends in that as to terminology. On the one hand, we observe, constant renovation of the names for, in the general-that, one and same things (certainly, technologies too develop, but rates of the change of their names much above). With other - we use the old terms for notion, sense which already quite not that earlier. Exactly, the second event exists to DBMS with reference to.

In explanatory dictionaries on computing machinery, released in 2002, happens to such determination a managerial system database (database management system): "exhibit, providing creation, keeping, renovation and information retrieval in database, as well as safety management and wholeness given". As a whole this interpretation was faithfully and 30 years back, but all profound part DBMS presently quite other, than in that distant timeses (we shall note that in determination already is absent the additional phrase, which was used for revision of the notion else eight years back, - "programme shell, being half way between database and user").

In the last decennial event we observe the situation, when DBMS changed from especially internal technological addenda to applied programme in independent product, around which are built applications for users; in other words, from one of the component of the information system - in platform for building of such systems.

In this situations cost(stand)s to pay attention the contents "platform Microsoft" to change. Traditionally operating system was meant under this term, Windows. However with reference to to server platform all more often we meet the ligament Windows Server + SQL Server. Moreover, introduces wholly real that with output at the beginning initially following year to new version Microsoft SQL Server (the worker name Yukon) we push with situation, when all rest products

Microsoft will be already written not under Windows, but under Yukon. Though exist and will exist the desk database: oddly to say, but Microsoft Access as judged by that as his(its) presents Microsoft - too DBMS.

Historically managerial system database were orientated on decision of the tasks, bound in the first place with transauctional processing to outline information. Certainly, best, time-tested decision here was and remains the relational model DBMS. However at the last years application database invariably enlarged. On the one hand, it is necessary to control the more broad set a format data, going to decision of the general problems of corporative information control. With other - exactly DBMS undertake the main functions to integrations data and applications of the corporative systems. (As of Gartner Group, information divisions enterprise spend before 40% its budget on decision of the tasks to integrations acting component database.) Exactly this are explained active interest to discussion architectral principle and possibilities to realization database different models – post relational, object-relational, XML.

If one tries to classify the existing application a database, but in the same way value the prospects of their development at present, that possible get the approximite list of the most wide-spread classes, got spreading and using in all application database. This list will look as follows:

- Documentographic and documentary are used in all base organ of power and control
- database on industrial, building and agricultural product
- database on economic and stated as affair to information (statistical, credit-financial, foreign trade)
- factographic bases social data, including information about population and about social ambience
- database of the transport system

- reference given for population and institutions (the encyclopedias and reference books, timetables plane and train, the address and telephones of the people and organization)
- resource database, including factographic information on natural resource (the land, water, depths, biological resources, weather forecast and waste, ecological situation)
- factographic bases and banks scientific data, providing fundamental scientific studies
- factographic database in the field of cultures and art
- linguistical database that is to say machine dictionary of the miscellaneous of the type and purposes.

The Economic tasks, for decision which necessary to use software DBMS, more extensive and varied. On his(its) base are built information systems enterprise different level (from small before large). The Applications database traditionally occupies that field of activity of the person, where he happens to to face the big volume to varied information. The First database were basically used in such fundamental science as, nucleus physics, chemistry, astronautics, and the other science requiring systematic approach to work with data. The most Further development computer technology and computerization society have brought about that that, database become be developed in all sphere of activity of the person practically, and aplying enterprise in miscellaneous from agriculture before financial-economic systems. The Last innovations of the using database became the worldwide web Internet, which in itself is an enormous database. Accordingly such spreading database requires and new software programmes of control them.

As it was mentioned above we can conclude that in order to run any sphere effectively using information technologies is considered to be a productive way. Moreover using database for working with information can provide effectiveness and safe time.

The aim of the database that we are going to do is to increase effectiveness. Mainly in this database topics of qualification papers will be kept. In order to do some tasks on it we are going to do a separate programme (interface).

Choosing qualification paper topics is one of the actual tasks that High Educatinal Institutions come across. In nowadays rapid progress of science and technology suitable qualification paper topics should be chosen. It is no use doing qualification thesis projects according to old topics. First of all before programming the database we should point what qualification paper topic is. The theme is the name of nowadays actual problems of science and technology. And the database of the present qualification paper deals with above mentioned problems, i.e. it is called problem base or problem bank. It is important to find out the fact that how they are added to the base.

The problems will be added to the base by the following sources:

- Scientific superviser
- Student
- Proffessional institution

We should find out the process of adding a problem to the base. For adding a problem to the base, first we need to study its structure.

A problem has the following structure:

- The name of the problem (further called theme),
- The author who adds a problem to the base,
- The direction of the problem,
- Key words,
- A short annotation of the problem.

So, the structure of the problem has been found out. Then we need to reveal in what language we will write the promlem theme.

The theme may be in Uzbek, Karakalpak, Russian or English. The database that we are going to do is planned to be used at Tashkent university of information technologies named after Muhammad Al-Khwarizmi Nukus branch. So we can

write a qualification paper and defend it in these languages. We can not choose only one of these languages, so the four language types will be written to the base. And one notable side is to prevent repeating themes. For example, an author presented a topic in the Russian language, and then another author presented this topic in the other language without knowing about that. As a result of that one topic would be written twice in two types. It follows to the repetition of the topics. Saving topics in different languages prevents that.

The next task is how to use the base after adding a problem. When the base is supplied with enough problems, they will be given to graduates. Graduates choose a topic that they are interested in, and then automatically the author i.e. scientific supervisor will be chosen.

If graduates have a topic for their qualification thesises that they are interested in then they can add them to the base. The database that we are going to do plays role at giving graduates actual themes i.e. for advice.

After choosing and defending a theme it will be added to the list of carried ones. Adding themes to the base will be done during the year. Runnig the base will be assigned to suitable users.

Working with the base will be done from one place only. That's why working by client-server method will not be necessary. The model of working with the database is shown in Figure 1-1. There is model IDEF0 drawn in the picture. It means that 3 different sources influence and result on the database. The object in the middle is the database. Entering information on the left is problems. Working mechanism from the top is user. The thing on the right side is result i.e. actual graduate work.

If the number of the information increases, various statistics can be made on. For example, the direction of the most used themes, the author whose themes are chosen mostly and others. It will make a lot of difficulties to do statistics according to simple method without using the database.

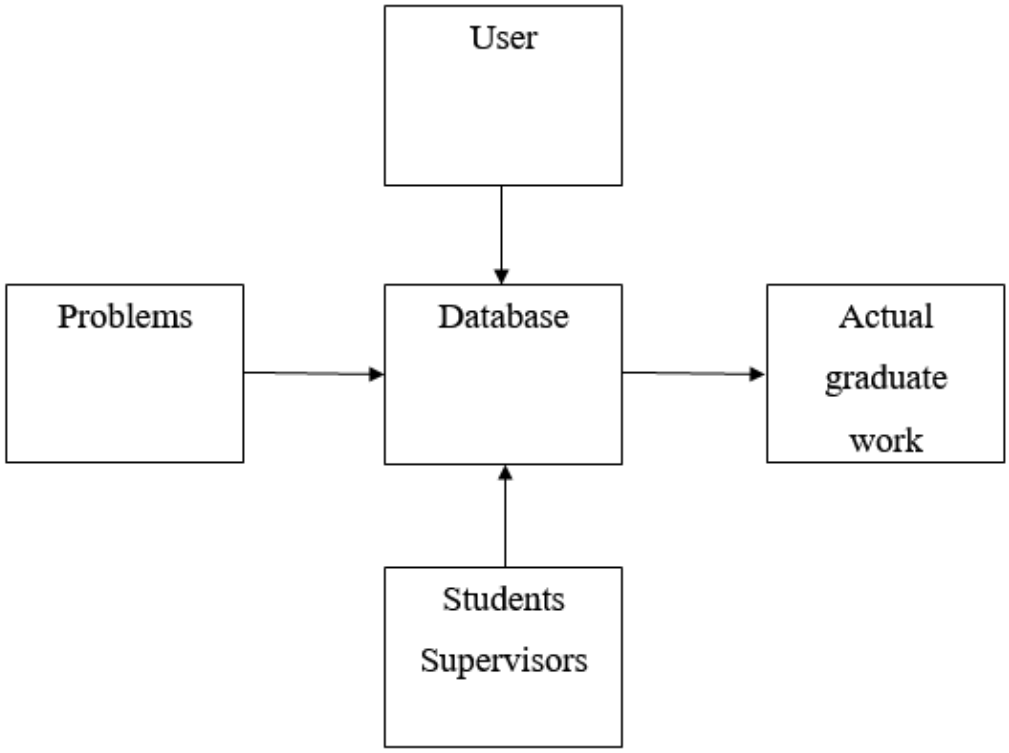

Figure 1-1. IDEF0 model of using database

1.2 Requirement to database

Correct designed Database must satisfy the following requirements:

1. Minimum redundancy. Consistency.
2. The Completeness of data.
3. The Independence of data.
4. The Possibility of conduct (the accompaniments and removing) and actualizations (the adjustments, modification) data.
5. Safety and secrecy.
6. High efficiency. The Minimum expenseses.
7. Observance standard.

1. Minimum redundancy means that given in Database must be not duplicated. If redundancy data exists, it draws two dangers:

- unjustified heavy expences to memories and reduction of time of the response of the system when processing the excessive greater volumes data.

- a breach consistency data i.e. origin to such situations, when in different places of the machine memory is kept inconsistent data. Origin of consistency is exceedingly dangerously for Database.

Contradictory can appear as a result of adjustments surplus data. When contributing the changes to logical record can happen so that separate copies this record, keeping in different places of the machine memory, turn out to be uncorrected. The Programmer happens to show emphases to organizations of the process of the adjustment surplus data and develop the special programs, preventing appearance of contradictory.

Contradictory can appear and at adjustment irredundant data. Site keeping data is a reason to high probability that that two or more user simultaneously are necessary one and same data. If one of the users addresses to data, but another in ditto time puts into them change, will are received inconsistent data. It Is Explained this that that process of the renovation data requires certain time, during which one and same given turn out to be on different stages of the renovation. When referencing to such given parallel working programs will are received inconsistent information

In DBMS exist the complex mechanisms of blocking updated given from access to him other users. The Parallel requests to one and same given are usually executed consecutively.

There is facility In row DBMS, preventing duplication and origin of consistency data. Otherwise, such facility develops the toolsmith

2. The completeness of data means that in Database must be kept only correct data i.e. they are kept logical conditions, in accordance with which given are considered correct. The Destruction and distortion given possible as a result unwary action of the users, as a result error in program and malfunctions of the equipment.

Exist the special methods and receiving the provision to wholeness. For provision of wholeness on data, keeping in Database, superimpose the restrictions. Are they herewith defined condition, which must correspond to importances data.

For instance, the same serving can not have two different years of the birth etc. The Similar restrictions are identified the law Database. Satisfiability of the laws Database is seasonly checked DBMS.

For prevention of the possibility of the entering wrong given is developed facility of the checking to correctness introduced data. So, for instance, possible use the procedures, checking attribute of introduced importances determined range of possible importances. For instance, amount of the workday is limited overhand by amount of the days at the current month.

Wholeness data can be broken under unchancy termination of the transactionses. The Transactions is identified certain неделимая operational procedure on data, executed on one request to DB. The Example to transactionses is an operation of the translation of the money with one count on another in bank system. Here necessary consequent execution several operations. The Money leave with one count, given are corrected, then money are added to the other count and given are newly corrected. If at least one of the actions is not executed successfully, result to transactionses will turn out to be invalid. DBMS must trace the move of the execution to transactionses from begin before her (its) terminations. If on some reason one of operation was not executed, that transactions is cancealled completely. Herewith "recoil" is executed by cancelling all already executed change.

Facility of the reconstruction given must be provided In DB after programme malfunctions and malfunctions of the equipment. Existing the program of the creation reserve mines and special programs, which automatically fix any contributed in DB change (s the file of the proofreadings). If the running version DB spoiled, that is the previous version, in she is contributed change, fixed in file of the proofreadings, and the current (actual) condition DB is restored.

Different DBMS in one or another measure dispose the facility of the provision to wholeness data. Otherwise such facility are developed by toolsmiths.

3. Independence data means that applied programs must not depend on prestored data i.e. from way of keeping given in physical memory. This allows to

add in DB new data, change the structures of keeping data, create on DB new applications. Earlier created program herewith must not "feel" these change.
DBMS usually provide this requirement.

4. The Structure DB must allow to include new and delete the outdated data, correct prestored given without destruction of the logical relationships, installed in scheme DB. For this scheme DB must be it is correct is designed, but operations of conduct DB must not break the scheme DB.

5. Safety and secrecy means protection given from unauthorized access, intentional and undeliberate destruction data, misappropriations data. The System of protection DB is called to solve the following tasks.

- An Identification of the users. The Data, keeping in DB must use only person, having on this right and confirmed their own authorities. The most wide-spread way of the decision of this task is a system of the passwords.
- Restriction of the access to data. Each user must work with that data only, which required for decision of his(its) tasks, rest data must be for it "unencountered.

Each user is given determined authorities (the privileges) for work with data. He can be given right only reading from DB, right of the entering in DB or right of the renovation etc. All privileges are given only manager DB

- Privacy protection data. Secret given necessary to protect from access by system special, it is enough complex passwords. Powerfully, the vulnerable data follows to encode.

The Meanses of protection and provision to safety data contains in DBMS or are developed by toolsmiths.

6. The Organization DB and access methods to data must provide the high velocity a data processing such that user could work with DB in dialogue modes. The Cost of the servicing the users must not be high. The Possibility of the execution of these requirements is defined beside factor: volume prestored data, speed of the technology, and way to organizations given in DB and in многом depends on decisions, taken developer in step of creation DB. For instance,

possible organize the location mode given on carrier by thereby that most often used given is kept on the most available area of the external memory.

7. The Presentation given in DB, accompanying documentation, way of the interaction of the user with DB must satisfy the certain standard. The Standards can be corporative, departmental, industrial, national and international. Observance standard absolutely required for joint use data and for organization of the exchange given between separate systems.

CHAPTER 2. DESIGNING DATABASE

2.1 The conception of database and DBMS

The Database (BD, database) - a named collection outline data, referring to determined application domain.The Application domain - certain part real existing systems, functionning as independent unit.The Full application domain can present itself economy of the country or groups union state, however in practice for information systems most importance has an application domain of the scale of the separate enterprise or corporations.

The Managerial system database (DBMS) - a complex programme and language facilities required for making and modification database, accompaniments, modification, removing, searching for and selection to information, presentations to information on screen and in printed type, delimitations of the rights of the access to information, execution other operation with the base.

Relational DB - a main type modern database. Consists of tables, between which can exist the relationship on key importances.

The Table database (table) - a regular structure, which consists of sister lines (record, records), divided into column (the field, fields).

In theories relational database synonym of the table - an attitude (relation), in which line is identified the tuple, but column - an attribute.

In conceptual model relational DB analogue of the table is essence (entity), with determined by set characteristic - an attribute, capable to take definite sign (the set of possible importances - domain).

The Key element of the table (the key, regular key) - such her(its) field (the idle time key) or строковое expression, formed from importances several flaps (the component key), on which possible define importances other flap for one or several records of the table. In practice indexes for use ключей - a page footing,

containing ranked information about key importances. In relational theory and conceptual model notion "key" is used for attribute relations or essence.

The Primary key (primary key) - a main key element, uniquely identifying line in table. Can also exist alternative (candidate key) and unique (unique key) keys, serving also for identification of the lines in table.

In relational theory primary key - a minimum set attribute, uniquely identifying tuple in respect of.In conceptual model primary key - a minimum set attribute to essence, uniquely identifying copy to essence.

The Relationship (relation) - a functional dependency between object. In relational database relationship are fixed between table on key, one of which in the main (parent, parental) to table - primary, the second - an external key - in external (child, affiliated) to table, as a rule, primary is not and forms the relationship "one to many" (1:N). In the event of primary external key relationship between table has a type "head-to-head" (1:1). Information on relationship is saved in database.The External key (foreign key) - a key element subordinated (external, affiliated) of the table, which importance complies with importance of the primary key main (parental) of the table.Reference wholeness given (referential integrity) - a set of the rules, providing correspondence to of key importances in bound table.

The Prestored procedures (stored procedures) - a programme modules, saved in database for performing determined operation with information of the base.

The Triggers (triggers) - prestored procedures, enforcing conditions to reference wholeness given in operation of the change primary ключей (possible cascade change given), removing the record in the main to table (cascade removing in affiliated table) and accompaniments of the record or change given in affiliated table.

The Object (object) - an element of the information system, possessing determined characteristic (properties) and determined by image ing on external events (events).

The System - a collection interacting between itself and with external encirclement object.

Replication database - a creation mines database (the remarks), which can be changed updated data or реплицированными by forms, report or the other object as a result of performing the process to synchronizing.

The Transactions - a change to information in the base as a result of execution one oring to their sequences, which must be executed completely or is not executed in general. The special mechanisms of the provision transactions exist In DBMS.

The Language SQL (Structured Query Language) - an universal language of the work with database, including possibility of her(its) creation, modification of the structure, selection given on request, modification to information in the base and other operations манипулирования database.

Null - importance of the field of the table, showing that information in given field is absent. Permit on possibility of existence of importance Null can be assigned for separate flap of the table.

Relational database

The Relational systems far from have immediately got broad spreading. While the main theoretical result in this area were received as far back as 70-h and then appeared the first prototypes relational DBMS, was for a long time considered impossible to obtain the efficient realization of such systems. However gradual accumulation of the methods and algorithm to organizations relational database and management they have brought about that that already in medium 80-h relational systems have practically displaced with world market early DBMS.

The Relational model given is founded on mathematical principle, resulting from theory of sets and logic predicate directly. These principles for the first time were aplying in the field of modeling given at the end 1960-h gg. doctor E. F. Koddom, in that time worked in IBM, but are for the first time published in 1970.

The Technical article "Relational model given for greater separated databanks of" doctor E. F. Codd, published in 1970, is an ancestor to modern theory relational DB. Doctor Codd has defined 13 rules to relational model (which name Codd’s 13 rules).

Codd's 13 rules

- Relational DBMS must be capable completely to control database through her (its) relational possibilities.
- Information rule - whole information in relational DB (including name of the tables and column) must be defined strictly as importances in table.
- Guaranteed access - any importance in relational DB must be гарантированно available to use through combination of the name of the table, importances of the primary key and name column.
- Support of empty importances (null value) - DBMS must know how to work with empty importances (unknown or unused importances), unlike importances by default and for any domains independently.
- Online relational catalogue - a description DB and her (its) contents must be submitted for logical level as tables, to which possible use requests, using language database.
- Comprehensive language of management given - at least once, one of supported languages must have clearly determined syntax and be all-embracing. He must support description of the structure data and manipulating them, rules to completeness, authorization and transactionses.
- Rule of the renovation of the presentations (views) - all presentations, theoretically updated, can be updated through system.
- Insertion, renovation and removing - DBMS supports not only inquiry for selection data, but also insertion, renovation and removing.
- Physical independence given - on program-applications and special programs logically do not affect change physical access method to data and structures vault data.
- Logical independence given - on program-applications and special programs logically do not affect, within reasonable, change the structures of the tables.

- Independence to wholeness - a language DB must be capable to define the rules to wholeness. They must be saved in an online reference book, and must not exist way their avoid.
- Independence of the distribution - on program-applications and special programs logically does not affect, first are once used data or again.
- Continuity - impossibility to avoid rules to wholeness, determined through language database, use the low-level languages.

Relational algebra

Main idea of the relational algebra consists in that that if soon relations are an ensemble, facility by manipulating relations can be based on traditional theorist-plural operation, complemented some special operation specific for relational database.

Exists much approaches to determination of the relational algebra, which differ the set an operation and way to their interpretation, but, in principle, are more or less tantamount. Extended initial variant of the algebra, which was offered Koddom, is identified algebra Kodda.

In this variant set main algebraic operation consists of eight operations, which are divided on two classes - a theorist-plural operation and special relational operations. In composition theorist-plural operation enter to operations:

- associations of the relations;
- crossing the relations;
- taking to differences of the relations;
- taking the cartesian making the relations.

Special relational operations include:

- restriction relations;
- projection relations;
- joining the relations;
- fission of the relations.

Besides, in composition of the algebra is included operation of the assignment, allowing save in database results calculations of the algebraic expressions, and operation of the renaming attribute, enabling correct to form headline (scheme) resulting relations.

When performing the operations of the association (UNION) two relations with alike headline is produced attitude, including all tuples, which enter at least in one of the relations - an operand.

Operation of the intersection (INTERSECT) two relations with alike headline produces attitude, including all tuples, which fall into both relations-operand.

Attitude, being difference (MINUS) two relations with alike headline, includes all tuples, falling into attitude - a first operand, such that none of they do not fall into attitude, which is a second operand.

When performing the Cartesian product (TIMES) two relations, intersection headline which emptily, is produced attitude, which tuples are produced by associations of the tuples first and second operand.

Result of the restriction (WHERE) relations on some condition are an attitude, including tuples relations-operand, satisfying this condition.

When performing the projections (PROJECT) relations on given subset ensemble his(its) attribute is produced attitude, which tuples are corresponding to subset of the tuples relations-operand.

When joining (JOIN) two relations on некоторому condition is formed resulting attitude, which tuples are produced by associations of the tuples first and second relations and satisfy this condition.

Beside operations of the relational fission (DIVIDE BY) two operands - binary and monadic relations. Resulting attitude consists of monadic tuples, including importances of the first attribute of the tuples of the first operand such that ensemble of importances of the second attribute (under fixed importance of the first attribute) includes ensemble of importances of the second operand.

Operation of the renaming (RENAME) produces attitude, which body complies with body of the operand, but name attribute are changed.

Operation of the assignment (:=) allows saving result of the calculation of the relational expression in existing attitude DB.

Kodd has offered using the relational algebra in DBMS, for dismemberment given in bounded sets. He has organized its system DB around concept, founded on set data.

In relational model given are split on sets, which form the tabular structure. This structure of the tables consists of the individual element data, named by fields. Single set or group by flap known as record.

Model data, or conceptual description of the application domain, - an most abstract level of the designing database.

With standpoint of the theories relational BD, the cardinal principles to relational model on conceptual level possible to formulate as follows:

- all given introduce in the manner of ranked structure, determined in the manner of lines and column and named by attitude;
- All importances are a scalar. This means that for any line and column any relations exists one and only one importance;
- All operations are executed on integer by attitude, and result of their execution also is an integer attitude. This principle is identified closing.

Formulating principles to relational model, doctor Kodd has chosen the term "attitude" (relation) since, on his(its) opinion, this term unambiguous (while, for instance, term "table" has an ensemble different type - a table in text, spreadsheet and pr.). The more wide-spread following error: relational model is named so therefore that she defines the relationship between tables. Indeed, name to this models derives from relations (tables database), liing in her (its) base.

Each line, containing data, is identified tuple, each column relations is identified attribute (at a rate of practical work with modern relational DB are used terms "record" and "field").

Element of the description to relational model given on conceptual level are essence, attributes, домены and relationship.

Essence - certain separate object or event, information on which necessary to save in database, having determined set characteristic - an attribute. Essence can be as physical (the real existing objects: for instance, STUDENT, attributes - a number зачетной books, surname, his(its) faculty, profession, number of the group and t. d.), so and abstract (for instance, EXAM, attributes - discipline, date, teacher, auditorium and pr.). Her(its) type and copy distinguish For essence. Type is characterized by name and property list, but copy - concrete importances characteristic.

Attributes to essence be:

Identifying and descriptive. The Identifying attributes have unique importance for essence given type and are a potential key. They allow uniquely recognizing the copies to essence. From potential keys is chosen one primary key (PC). As PC is usually chosen potential key, on which more often occurs address to copy record. PC must comprise of its composition minimum necessities for identification amount attribute. Rest attributes are identified descriptive.

Simple and component. The Simple attribute consists of one component, his (its) importance indivisible. The Component attribute is a combination several components, possible, belonging to different types given (for instance, the address). Decision on that, use component attribute or split him(it) on components, depends on particularities of the processes of his(its) use and can be connected with ensuring the high velocity of the work with greater database.

Unambiguous and ambiguous - can have accordingly one or much importances for each copy of essence.

The Main and derived. Importance of the main attribute does not depend on the other attribute. Importance of the derived attribute is calculated on base of importances other attribute (for instance, age of the person is calculated on base of the date of his (its) birth and current date).

Specification of the attribute consists of his (its) names, instructions of the type data and descriptions of the constraints - an ensemble of importances (or домена), which can take givenned attribute.

Domain - a set of all possible importances, which can contain the attribute. The Notion "domain" often muddle with notion "type given". Necessary to distinguish these two notions. The Type given - a physical concept, but domain - logical. For instance, "integer number" - a type data, but "age" - domain.

The Relationship - on conceptual level present itself simple assotiations between essences. For instance, statement "Buyers gain the products" indicates that between essence "Buyers" and "Products" exists the relationship, and such essence are identified the participant this relationship.

Exists several types of the relationships between two essences: this relationship "one-to-one", "one-to-many" and "many-to-many".

Each relationship in relational model is characterized by name, обязательностью, type and degree. Distinguish optional and obligatory relationship. If essence of one type turns out to be perforce to be connected with essence of the other type then between these types object exists the obligatory relationship (double line is marked). Otherwise relationship is optional.

The Degree relationship is defined by amount of essence, which are engulfed given by relationship. Example binary relationship - a relationship between division and employee, which works in.

2.2 Designing database

One of the important stages at development information system this designing database. Primary tasks of the designing database:

- Provision of keeping in DB whole necessary information.
- Ensuring the possibility of the reception given on all necessary requests.
- Reduction to redundancy and duplication data.
- Provision to wholeness database.

Stages designing database:

- Conceptual (infological) designing
- Logical (datological) designing
- Physical designing

Conceptual (infological) designing

Conceptual (infological) designing - a building to semantic model of the application domain that is to say information model the highest level to abstractions. Such models without orientation on some concrete DBMS and model data. The Terms "semantic model", "conceptual model" and "infological model" are a synonym. Besides, in this context equal can be used word "model database" and "model of the application domain" (for instance, "conceptual model database" and "conceptual model of the application domain") since such model is as image to realities, so and image designed database for this realities.

The Concrete type and contents to conceptual model database is defined chosen for this formal device. Are they usually used graphic notations, like ER-diagram?

Most often, the conceptual model database comprises of itself:

- Description information object or notion of the application domain and relationships between them.
- Description of the constraints i.e. requirements to possible importances data and to relationship between them.

Logical (datelogical) designing

Logical (datelogical) designing - a making the scheme database on base of the concrete model data, for instance, relational model data. For relational model given даталогическая model - a set of the schemes of the relations, with instruction primary keys usually, as well as "relationships" between relations, presenting itself external keys.

The Transformation to conceptual model in logical model, as a rule, is realized on formal rule. This stage can be is to a considerable extent automated.

In step of logical designing is taken into account specifics to concrete model data, but can be not taken into account specifics concrete DBMS.

Physical designing

Physical designing - a making the scheme database for concrete DBMS. Specifics concrete DBMS can comprise of itself restrictions on именование object database, restrictions on supported types given etc. Besides, specifics concrete DBMS under physical designing includes choice of the decisions, in accordance with physical ambience of keeping given (choice of the methods of disc memory control, division DB on file and device, access method to given), creation index and etc.

Models «essence-relationship»

The Model "essence-relationship" (the angel. "Entity-Relationshipmodel"), or ER-model, offered by P. CHenom in 1976, is a most known representative of the class semantic (conceptual, infological) of the models of the application domain. ER-model usually introduces in graphic form, with use the original notation P. CHena, named ER-diagram, or with uses other graphic notation (Crow's Foot, Information Engineering and others.).

Main advantage ER-models:

- clarity;
- models allow to design database with big amount object and attribute;

ER-models marketed in many system computer aided design database (for instance, ERWin).

Main elements ER-models:

- objects (essence);
- object tools;
- relationship between object.

Essence - an object of the application domain, havinging attributes.

Relationship between essences is characterized:

- type relationship (1:1, 1:N, N:M);

- Class accesories. The Class can be obligatory and unnecessary. If each copy to essence participates in relationship, that class accesories - obligatory, otherwise - unnecessary.

Semantic models

The Semantic model (the conceptual model, infological model) - a model of the application domain, intended for presentation of the semantics of the application domain on the most high level of the abstractions. This means that is eliminated or minimized need to use notions "low-level", connected with specifics of the physical presentation and keeping data.

Normalization data

The Important stage when designing database this normalization data that is to say brings the tables a database on normal form. The Normal form - a characteristic relations in relational model data, characterizing him(it) with standpoint of redundancy, potentially bring about logically wrong result of the sample or change data. Normal form is defined as collection of the requirements, which must satisfy attitude.

The Process of the transformation of the relations database to type, answerring normal forms, is identified the normalization. The Normalization is intended for adduction of the structure DB to type, providing minimum logical redundancy, and does not aim the reduction or increase to capacity of the work or reduction or increase the physical volume database. The Long-run objective to normalizations is a reduction potential consistency prestored in database of information. General-purpose of the process to normalizations is concluded in following:

- exception of some types to redundancy;
- removal some anomaly renovations;
- development of the project database, which is it is enough "qualitative" presentation of the real world, intuitive comprehensible and can serve good central to following expansion;

- simplification of the procedure of the using the necessary constraints.

Removal to redundancy is produced, as a rule, for count of the decompositions of the relations by so as in each attitude were kept only primary facts (that is to say facts, not taken out from other prestored fact).

First normal form (1NF)

Variable relations is found in first normal form (1НФ) iff, when in any possible importance relations each his(its) tuple contains only one importance for each of attribute.

In relational model attitude is always found in the first normal form on determination of the notion attitude. That concerns different tables, that they can be not correct presentations of the relations and, accordingly, can be not in 1НФ.

Second normal form (2NF)

Variable relations is found in second normal form iff, when she is found in first normal form and each not key attribute неприводимо (function packed) depends on her(its) potential key.

Third normal form (3NF)

Variable relations is found in third normal form iff, when she is found in second normal form, and are absent transitive to functional dependencies not key attribute from key.

The first task of planning database is to gather essential information. The information which is kept in the present qualification thesis is analysed in the first chapter and according to it we will plan the database.

The following information is kept in the database «Problem base»:

- The theme of the qualification thesis
    - Uzbek version of the theme
    - Karakalpak version of the theme
    - Russian version of the theme
    - English version of the theme
- Category of the theme

- The information about the author
    - Name of the author
    - The name of the department where he works
    - Author's position in the department
    - Scientific degree of the author
- Added date to the base
- Key words connected to the theme
- Annotation
- The information about the theme defence
    - Name of carried student
    - Scientific superviser
    - The date of defencing

The database will be planned according to the information mentioned above. Normal rules were used while planning. That's why doing separate tables for category, department, position, degree and authors is important. And these tables will be logically connected with the tables of main themes. The table connected to the main tables is saved as an ID number. Here the repeated information is replaced by numbers. As a result the volume of the database will be shortened a little bit. The tables of planning informatin and their relationship are shown in figure 2-1.

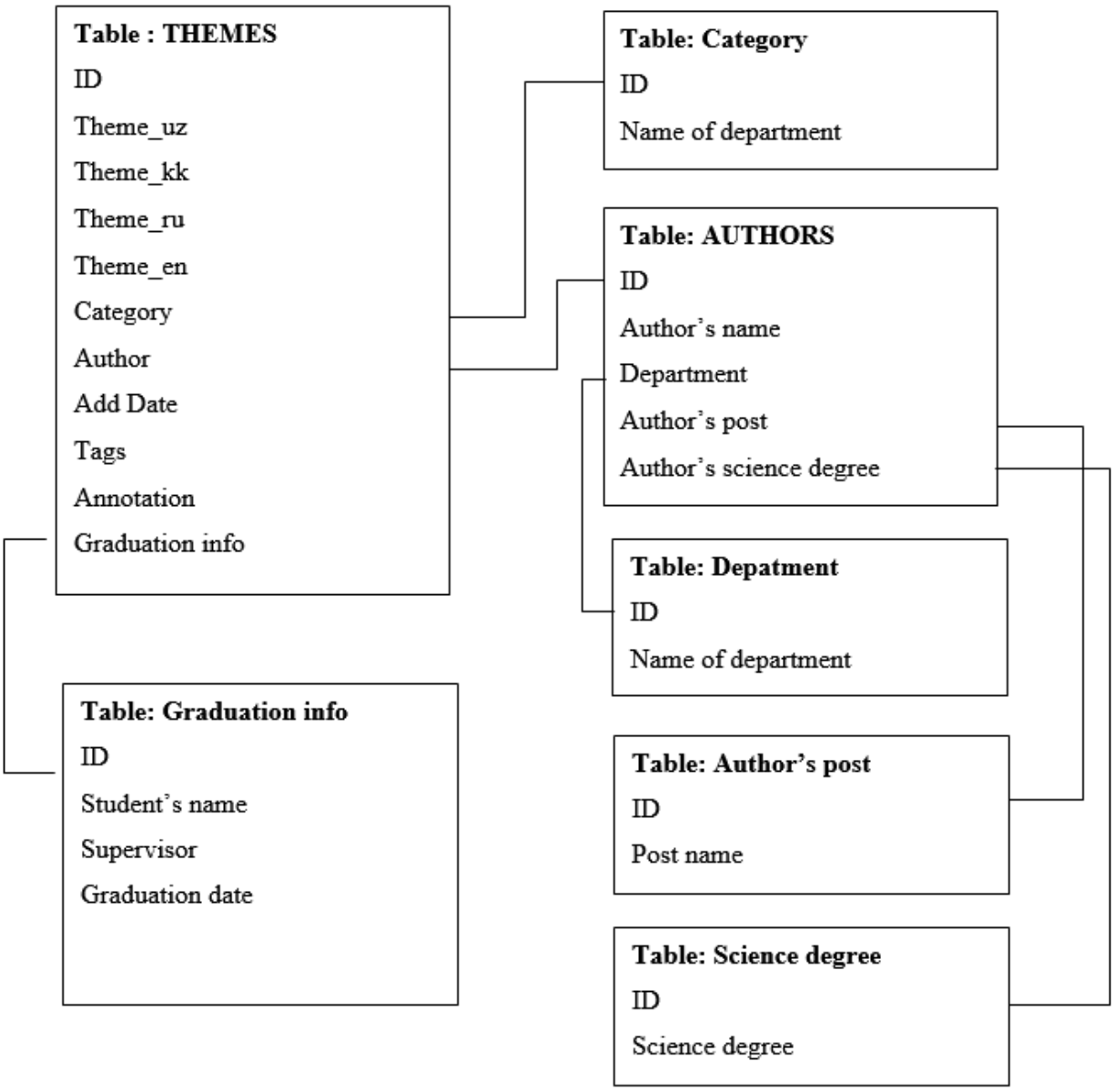

Figure 2-1. The list of database tables.

CHAPTER 3. PROGRAMM IMPLEMENTATION OF THE DATABASE AND INTERFACE

3.1 Selection of the necessary tools

The embedded database SQLite

SQLite History

SQLite's author is D. Richard Hipp. The program originally ran on Hewlett-Packard Unix (HPUX) and used an Informix database as the back-end. For their particular application, Informix was somewhat overkill. For an experienced database administrator (DBA), it could take almost an entire day to install or upgrade. To the uninitiated application programmer, it might take forever. What was really needed was a self-contained database that was easy to use and that could travel with the program and run anywhere regardless of what other software was or wasn't installed
on the system.

In January 2000, Hipp and a colleague discussed the idea of creating a simple embedded SQL database that would use the GNU DBM B-Tree library (gdbm) as a back-end, one that would require no installation or administrative support whatsoever. Later, when some free time opened up, Hipp started work on the project, and in August 2000, SQLite 1.0 was released. As planned, SQLite 1.0 used gdbm as its storage manager. However, Hipp soon replaced it with his own B-tree implementation that supported transactions and stored records in key order. With the first major upgrade in hand, SQLite began a steady evolution, growing in both features and users. By mid-2001 many projects—both open source and commercial alike— started to use it. In the years that followed, other members of the open source community started to write SQLite extensions for their favorite scripting languages and libraries. One by one, new extensions—an Open Database Connectivity (ODBC) interface followed by extensions for Perl, Python, Ruby,

Java and other mainstays—fell into place and testified to SQLite's wide application and utility.

SQLite began a major upgrade from version 2 to 3 in 2004. Its primary goal was enhanced internationalization supporting UTF-8 and UTF-16 text as well as user-defined text-collating sequences. While 3.0 was originally slated for release in summer 2005, America Online provided the necessary funding to see that it was completed by July 2004. Besides internationalization, version 3 brought many other new features such as a revamped C API, a more compact format for database files (a 25 percent size reduction), manifest typing, Binary Large Object (BLOB) support, 64-bit ROWIDs, autovacuum, and improved concurrency. In spite of the many new features, the overall library footprint was still less than 240 kilobytes. Another improvement in version 3 was a good code cleanup—revisiting and rewriting, or otherwise throwing out extraneous stuff accumulated in the 2.*x* series. SQLite continues to grow feature-wise while still remaining true to its initial design goals:

- Simplicity
- Flexibility
- Compactness
- Speed

Embedded Database

SQLite is an embedded database. Rather than running independently as a standalone process, it symbiotically coexists inside the application it serves—within its process space. Its code is intertwined, or embedded, as a part of the program that hosts it. To an outside observer, it would never be apparent that such a program had a relational database management system (RDBMS) on board. The program would just do its job and manage its data somehow, making no fanfare about how it went about doing so. But inside, there is a complete, self-contained database engine at work.

One advantage of having a database server inside your program is that no network configuration or administration is required. Both client and server run together in the same process. This reduces overhead related to network calls, simplifies database administration, and makes it easier to deploy your application. Everything you need is compiled right into your program.

Consider the processes found in Figure 3-1. One is a Perl script, another is a standard C/C++ program, and the other is an Apache process with PHP, all using SQLite. The Perl script imports the DBI::SQLite module, which in turn is linked to the SQLite C API, pulling in the SQLite library. The PHP library works similarly, as does the C++ program. Ultimately, all three processes interface with the SQLite C API. All three therefore have SQLite embedded in their process spaces, and all three are independent database servers in and of themselves. Furthermore, even though each process represents an independent server, they can still operate on the same database file(s), as SQLite uses the operating system to manage synchronization and locking.

For programmers, SQLite is like digital duct tape, providing an easy way to bind applications and their data. Like duct tape, there is no end to its potential uses. In a web environment, SQLite can help with managing complex session information. Rather than serializing session data into one big blob, individual pieces can be selectively written to and read from individual session databases.
SQLite also serves as a good stand-in relationaldatabase for development and testing: there are no external RDBMSs or networking to configure,or usernames and passwords to bother with. SQLite might also serve as a cache, hold configurationdata, or because of its binary compatibility across platforms, even work as an applicationfile format.

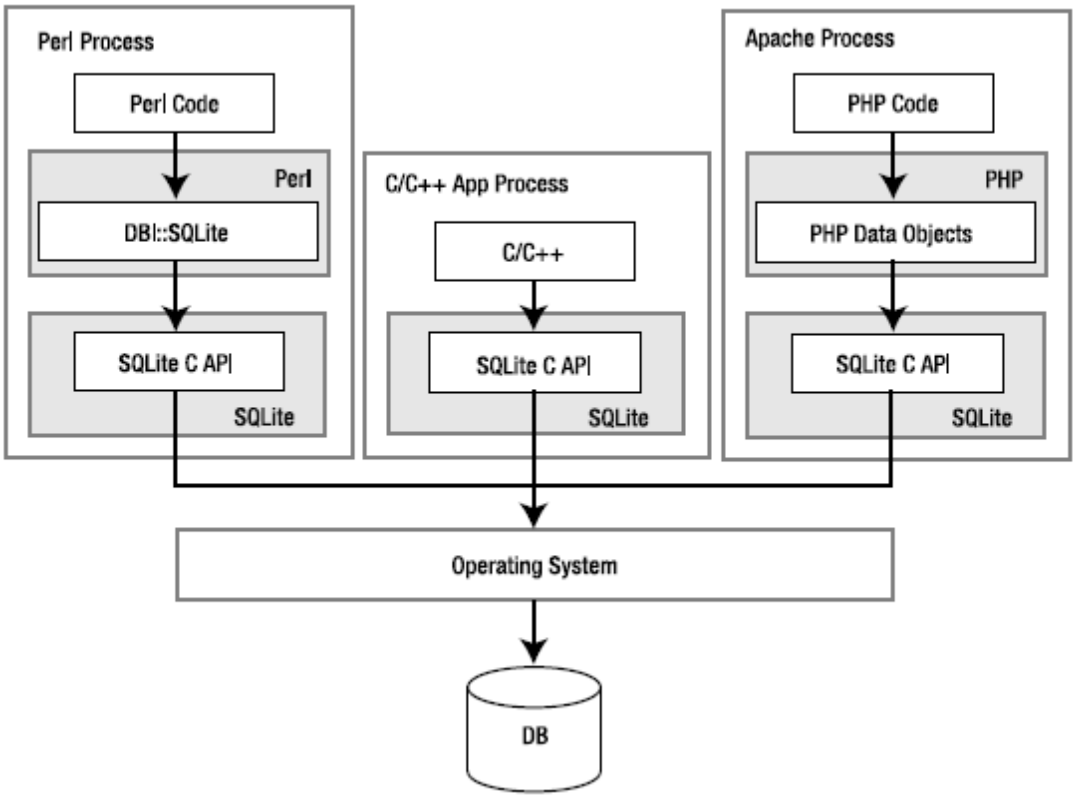

Figure 3-1. SQLite embedded in host processes

Besides being just a storage receptacle, SQLite can serve as a purely functional tool as wellfor general data processing. Depending on size and complexity, it may be easier to representsome application data structures as a table or tables in an in-memory database. This way, youcan operate on the data relationally, using SQLite to do the heavy lifting rather than having to
write your own algorithms to manipulate and sort data structures. If you are a programmer, imagine how much code it would take to implement the following SQL statement in your program:

```
SELECT AVG(z-y) FROM table GROUP BY x
HAVING x > MIN(z) OR x < MAX(y)
ORDER BY y DESC LIMIT 10 OFFSET 3;
```

If you are already familiar with SQL, imagine coding the equivalent of a subquery, compound query, GROUP BY clause or multiway join—in C. SQLite embeds all of this functionality into your application with minimal cost. With a database engine integrated directly into your code, you can begin to think of SQL

as a domain-specific language in which to implement complex sorting algorithms in your program. This approach becomes more appealing as the size of your data set grows or as your algorithms become more complex. What's more, SQLite can be configured to use a fixed amount of RAM and then offload data to disk if it exceeds the specified limit. This is even harder to do if you write your own algorithms. With SQLite, this limit is instituted with a single SQL command.

SQLite is also a great learning tool for programmers—a cornucopia for studying computer science topics. From parser generators, tokenizers, virtual machines, B-tree algorithms, caching, program architecture, and more, it is a fantastic vehicle for exploration of many well-established computer science concepts. Its modularity, small size, and simplicity make it easy to present each topic as an isolated case study that any one individual could easily follow.

But SQLite is not just a programmer's database. It is a useful tool for system administrators as well. It is small, compact, and elegant like a regular expression or a Unix utility such as find, rsync, or grep. SQLite has a command-line utility that can be used within shell scripts. However, it works even better with a large variety of scripting languages such as Perl, Python, and Ruby. Together the two can help with a wide variety of tasks, such as aggregating log file data, monitoring disk quotas, or performing bandwidth accounting in stateful firewalls. Furthermore, since SQLite databases are ordinary operating system files, they are easy to work with, transport, and back up. Also, SQLite is a convenient learning tool. It is an ideal beginner's database with which to learn about relational concepts. It can be installed quickly and easily on almost any platform, and its database files share freely between them without the need for conversion. It is full featured but not daunting. And it—both the program and the database—can be carried around on a floppy disk or Universal Serial Bus (USB) stick.

Unlike most RDBMS products, SQLite does not have a client/server architecture. Mostlarge-scale database systems have a large server package that makes up the databaseengine. The database server often consists of multiple processes that work in concertto manage client connections, file I/O, caches, query

optimization, and query processing.A database instance typically consists of a large number of files organized into oneor more directory trees on the server filesystem. In order to access the database, all ofthe files must be present and correct. This can make it somewhat difficult to move orreliably back up a database instance.

All of these components require resources and support from the host computer. Bestpractices also dictate that the host system be configured with dedicated service-useraccounts, startup scripts, and dedicated storage, making the database server a veryintrusive piece of software. For this reason, and for performance concerns, it is customaryto dedicate a host computer system solely for the database server software.

To access the database, client software libraries are typically provided by the databasevendor. These libraries must be integrated into any client application that wishes toaccess the database server. These client libraries provide APIs to find and connect tothe database server, as well as set up and execute database queries and commands. Figure 3-2 shows how everything fits together in a typical client/server RDBMS.

In contrast, SQLite has no separate server. The entire database engine is integrated into whatever application needs to access a database. The only shared resource among applications is the single database file as it sits on disk. If you need to move or back up the database, you can simply copy the file. Figure 3-3 shows the SQLite infrastructure.

SQLite's Features

SQLite offers a surprising range of features and capabilities despite its small size. It supports a large subset of ANSI SQL92 (transactions, views, check constraints, correlated subqueries, and compound queries) along with many other features found in relational databases, such as triggers, indexes, autoincrement columns, and the LIMIT/OFFSET clause. It also has many uniquefeatures, such as in-memory databases, dynamic typing, and something called conflict resolution (explained in a moment).

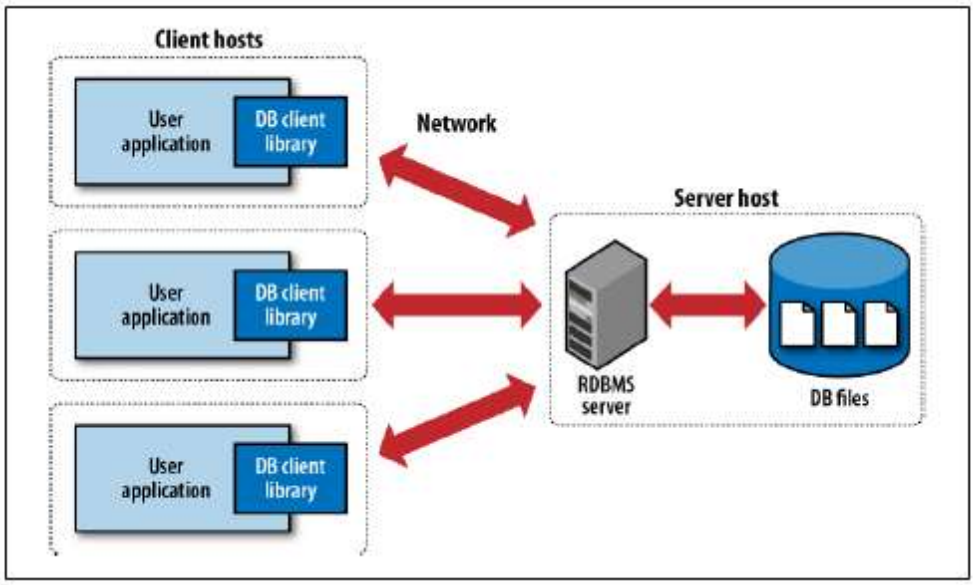

Figure 3-2. Traditional RDBMS client/server architecture.

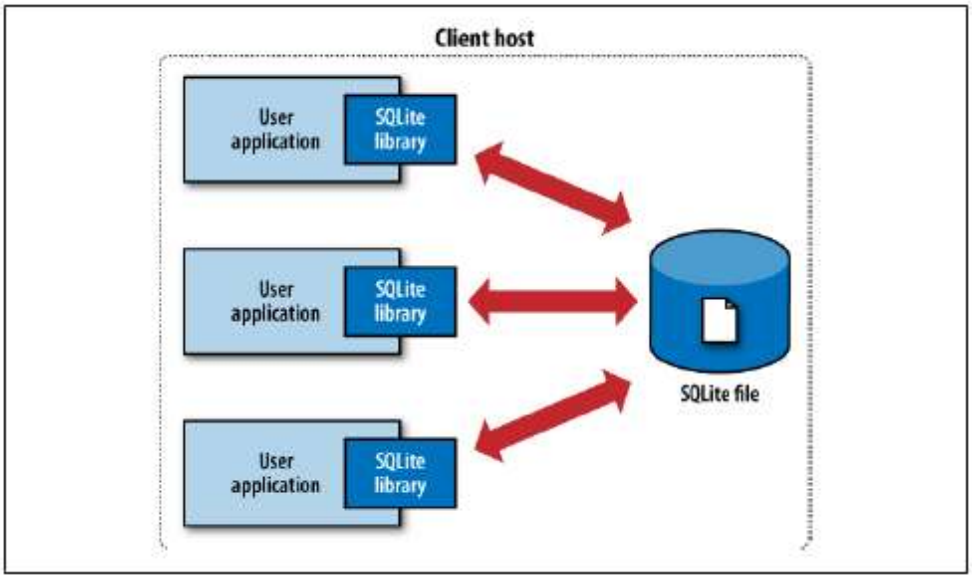

Figure 3-3. The SQLite server-less architecture.

Zero Configuration

From its initial conception, SQLite has been designed with the specific absence of a DBA in mind. Configuring and administering SQLite is as simple as it gets. SQLite contains just enough features to fit in a single programmer's brain, and like its library, requires as small a footprint in the gray matter as it does in RAM.

Portability

SQLite was designed specifically with portability in mind. It compiles and runs on Windows, Linux, BSD, Mac OS X, commercial Unix systems such as Solaris, HPUX, and AIX, as well as many embedded platforms such as QNX, VxWorks, Symbian, Palm OS, and Windows CE. It works seamlessly on 16-, 32-, and 64-bit architectures with both big and little endian byte orders. Portability doesn't stop with the software either: SQLite's database files are as portable as its code. The database file format is binary compatible across all supported operating systems, hardware architectures, and byte orders.

Compactness

SQLite was designed to be lightweight and self-contained: one header file, one library, and you're relational, no external database server required. Everything—client, server, virtual machine— packs into a tidy quarter megabyte, which at the moment is smaller than the home page of the publishers of this book: www.apress.com (the home page weighing around 260 kilobytes). Furthermore, there is a proprietary version of SQLite that is as small is 69 kilobytes, capable of running on smart cards (see the "Additional Information" section for more details). Equally compact are SQLite databases. They are ordinary operating system files. Regardless of your system, all objects in your SQLite database—tables, triggers, schema, indexes, and views—are contained in a single operating system file. Furthermore, SQLite uses variable-length records, allocating only the minimum amount of data needed to hold each field. A 2-byte field sitting in a varchar(100) column only takes up 3 bytes of space, not 100 (the extra byte is used to record its type information).

Simplicity

As a programming library, SQLite's API is one of the simplest and easiest to use. The API is both well documented and intuitive. It is designed to help you customize SQLite in many ways, such as implementing your own custom SQL functions in C. Better yet, the open source community has a created a vast number of language and library interfaces with which to use SQLite. Thereare extensions for Perl, Python, Ruby, Tcl/Tk, Java, PHP, Visual Basic, ODBC, Delphi, Microsoft

.NET, Smalltalk, Ada, Objective C, Eiffel, Rexx, Lisp, Scheme, Lua, Pike, Objective Camel, Qt, WxWindows, REALBASIC, and others. SQLite has a modular design. This design includes many innovative ideasthat enable it to be full featured and extensible while at the same time retaining a great degreeof simplicity throughout its code base. Each module is a specialized, independent system thatperforms a specific task. This modularity makes it much easier to develop each system independently,and to debug queries as they pass from one module to the next—from compilation andplanning to execution and materialization. Several factors work together to make SQLite a very flexible database. As an embedded database,it offers the best of both worlds: the power and flexibility of a relational database front-end,with the simplicity and compactness of a B-tree back-end. With it, there are no large databaseservers to configure, no networking or connectivity problems to worry about, no platform limitations,no baroque APIs to learn, and no license fees or royalties to pay.

Reliability

But the source code is more than just free.It also happens to be well written. SQLite's code baseconsists of about 30,000 lines of standard ANSI C, which is clean, modular, and well commented.It is designed to be approachable, easy to understand, easy to customize, and generally veryaccessible. It is easily within the ability of a competent C programmer to follow any part ofSQLite or the whole of it with sufficient time and study.Additionally, SQLite's code offers a full-featured API specifically for customizing andextending SQLite through the addition of user-defined functions, aggregates, and collatingsequences along with support for operational security.While SQLite's modular design significantly contributes to its overall reliability, itssource code is also well tested. Whereas the core software (library and utilities) consists ofabout 30,000 lines of code, the distribution also includes an extensive test suite consisting ofover 30,000 lines of regression test code, which covers over 97 percent of the core code. That is,over half of the SQLite project's total code is devoted exclusively to regression testing.

Anotherway of saying this is for every line of database code written, there is approximately one line oftest code written as well.

C# and .NET Platform

.NET Platform.

The Platform Microsoft .NET (and connected with her programming language With#) was for the first time presented approximately in 2002 and quickly became one of the main modern ambiences of the software development. As it was noted in introductory section of this book, under her(its) writing were pursued two purposes. The First of them - a granting reader deep and detailed description of syntax and semanticses of the language With#. The Second (so important) purpose - an illustration of the using multiple API-interface .NET, including access to database by means of ADO.NET and Entity Framework (EF), set technology LINQ, WPF, WCF, WF and development веб-site with use ASP.NET..

Before that, as company Microsoft has released the language C# and platform .NET, developers of software, created exhibits for operating systems family Windows, often used the programming model a COM. Technology COM (Component Object Model - a model компонентных object) allowed to build the libraries, which possible was use in different programming languages. For instance, programmer on C++ could build the library a COM, which could use the developer on Visual Basic. Nezavisimaya from language nature COM, certainly, was suitable. However defect to models COM was shown complicated infrastructure, frail model of the deployment and possibility of the work under management Windows only.

In spite of difficulty and restrictions COM, with using of this architecture was is successfully created literally uncountable amount of exhibits. However, at present majority of exhibits, oriented on family of the operating systems Windows, the models COM without using. Instead of this exhibits for desk computer, веб-

sites, services of the operating system and libraries repeatedly used logic of the access to data and business-logic are built by means of platforms .NET.

Advantage of the .NET platform.

As was it already said, language C# and platform .NET were for the first time presented in 2002, and purpose of their creation was a provision more powerful, flexible and model idle time of the programming in contrast with COM. As will possible see in rest material of the book, .NET Framework - a programme platform for building of exhibits on the base family operating systems Windows, as well as multiple operating systems production not Microsoft such as Mac OS X and different distribution programs Unix and Linux. For begin is below brought short list of some key facilities, supported by .NET.

• *Possibility of the interaction with existing code*. This possibility, certainly, is very useful since allows to combine the existing binary components a COM (i.e. provide the interaction with them) with more new programme component .NET conversely. With output .NET 4.0 and following version possibility interactions was in addition simplified due to accompaniment of the keyword dynamic.

• *Support of the multiple programming languages*. Exhibits .NET можносоздавать with use of any number of the programming languages (With#. Visual Basic, F# and etc).

• *The General acting mechanism, separated all supporting* .NET languages. One of the aspect of this mechanism is presence well determined set of the types, which capable to understand each maintaining .NET language.

• *The Language integration*. In .NET is supported inlanguaging inheritance, inlanguaging processing the exceptions and inlanguaging debugging the code. For instance, base class can be determined in With#, but then extended in Visual Basic.

• *The Extensive library of the base classes*. This library allows to avoid difficulties, in accordance with performing the low-level address to API-interface, and offers coordinated object model, used all supporting .NET languages.

• *The Simplified model of the deployment*. Unlike COM, the libraries .NET are not registered in system roll. Moreover, platform .NET allows to coexist on one and same computer several versions one and same assemblies *.dll.

Programming language C#

Syntax of the programming language C# looks much look like syntax of the language Java. However name C# with clon Java wrong. Actually and With#, and Java are a member family programming languages, founded on With (where also enter C, Objective C, C++), and therefore they separate similar syntax.

Many syntax designs C# simulated on base varied aspect languages Visual Basic (VB) and C++. For instance, like VB, language C# supports the idea a characteristic class (as opposition traditional method extractions and installation) and unnecessary parameters. Like C++, the language C# allows to overload to operations, as well as create structures, enumerations and functions of the inverse call (by means of delegate).

Moreover, on measure of the study of the material of this book, you much soon see that C# supports the ensemble of the facilities, which traditionally meet in different languages of the functional programming (for instance, LISP or Haskell), shall say, лямбда- expressions and anonymous types. In addition, with appearance of technologies LINQ (Language Integrated Query - a language integrated request), language C# became to support to designs, which do him(it) rather unique in the world of the programming. For all that, most influence upon he has rendered exactly languages, founded on C.

In consequence of that that C# presents itself hybrid from several languages, he is such syntax clear - unless clearer - either as Java, nearly so simple, as VB, and practically such powerful and flexible, as C++. Below full list of the key particularities of the language is brought far from With#, which typical of all his(its) version.

- Pointers to use is not required! In program on C# usually does not arise need for handling pointer stright (though in the event of absolute need be possible lowered on this level.
- Autocontrol by memory by means of assembly of the rubbish. Considering this, keyword delete in C# is not supported.
- Formal syntax designs for classes, interface, structures, enumerations and delegate.
- Similar language C++ possibility of the overloading operation for special types without excessive difficulties (for instance, provision "return *this, in order to let collect in chain" - not your care).
- Programming Support on base attribute. This variety of the development allows annotating types and their members for the reason additional revision of their behaviour. For instance if mark method by attribute [Obsolete], in an effort use of this member programmers gets corresponding to special warning.

All this leads us to the running version with#, being included in platforms .NET 4.5. The pair of the new keywords appeared in this versions C# (async and wait), which vastly simplify all-round and anisochronous programming. That, to whom happened to to work with preceding version With#, can remind that calling the methods through secondary flows required rather big volume of the difficult to understand code and use different space names .NET. Due to the fact that now C# supports language keywords, which automatically solve these tasks, process of the calling the methods by anisochronous image nearly so forgive; pardon, as their call in synchronous manner.

3.2 Database implementation

The next step after planning the database is to produce the database itself.

In the second chapter we have determined the informatin which will be sited in the database and divided them into tables. The next step is to choose a suitable database for our condition. Taking to consideration the demand to database i.e. taking to consideration single user, the comfortable database for us is SQLite.

This is the database which has been arranged. The necessary information for the database has been found out and it has been created.

Db Browser for SQLite programme was used for creating the database.

(Figure 3.4). The database was called MyDb.sqlite.

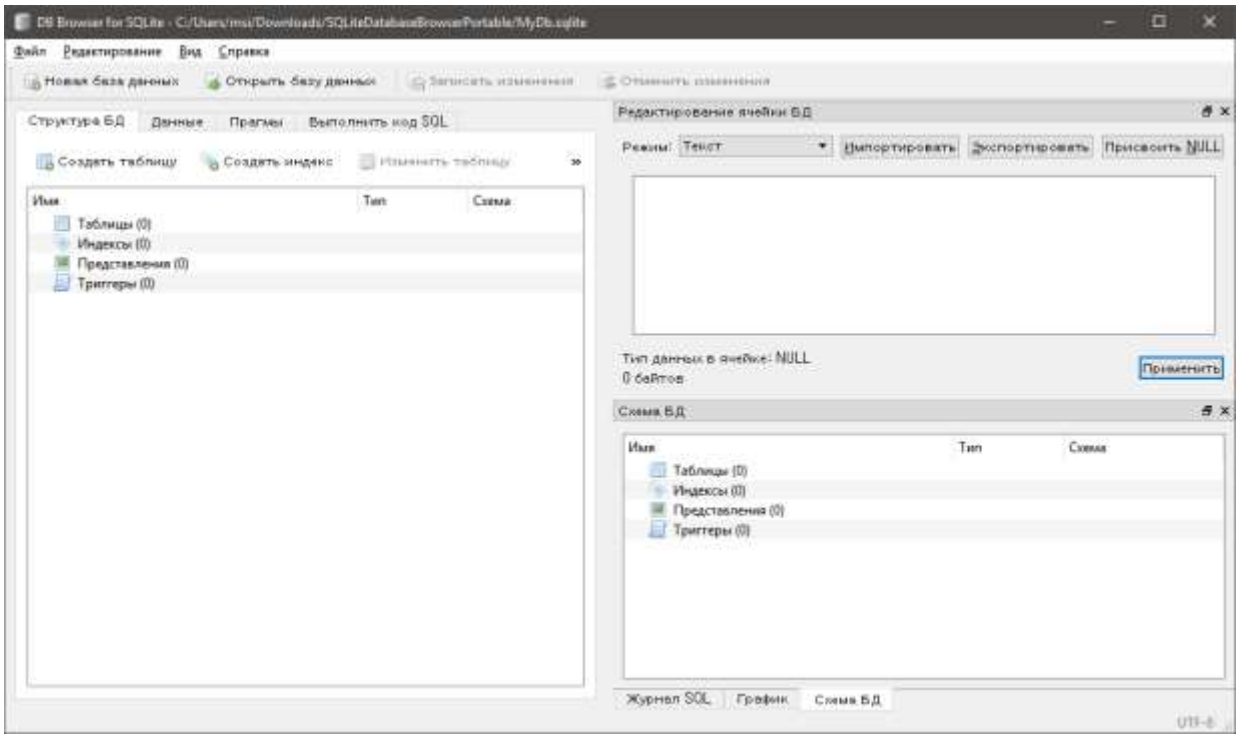

Figure 3-4. DB Browser for SQLite

After creating the base the tables have been created. We have used SQLiteStudio 3.1.1 programme while creating the tables. The interface of the programm is shown in Figure 3-5.

Having a lot advantages this resource has been chosen among other analogue resources as a programme.

To create tables in this programme, to fill them with information may be done by the method of graph or according to the rules of SQL.

And also this programming object has opportunities of seeing information, deleting or correcting it, importing-exporting tables and so on.

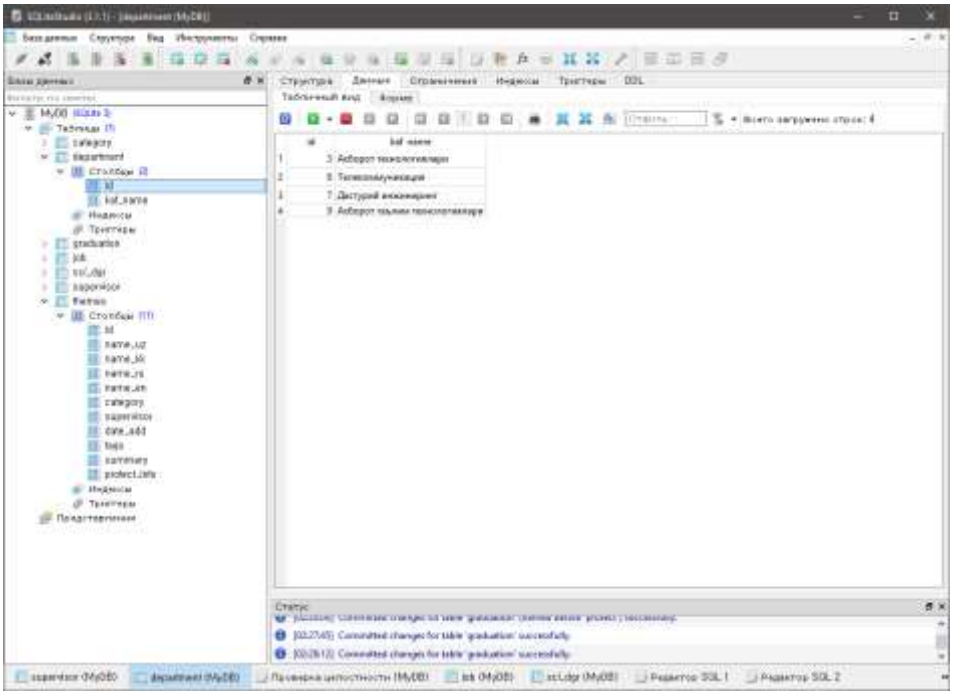

Figure 3-5. The interface of SQLiteStudio 3.1.1.

The first table is the theme table which saves the following themes:

```
CREATE TABLE themes (
id INTEGER PRIMARY KEY AUTOINCREMENT,
name_uz VARCHAR UNIQUE ON CONFLICT IGNORE,
name_kk VARCHAR,
name_ru VARCHAR,
name_en VARCHAR,
category INTEGER,
supervisor INTEGER,
date_add DATE,
tags TEXT,
summary TEXT,
protect_info CHAR (5) DEFAULT FALSE
);
```

The next table is the superviser table which saves the information about authors:

```
CREATE TABLE supervisor (
id INTEGER PRIMARY KEY,
name CHAR (40) UNIQUE ON CONFLICT IGNORE,
department INTEGER,
job INTEGER,
sci_dgr INTEGER
);
```

The category table which sets the directions of themes:

```
CREATE TABLE category (
id INTEGER PRIMARY KEY AUTOINCREMENT,
cat_name VARCHAR
);
```

Department table:

```
CREATE TABLE department (
id INTEGER PRIMARY KEY AUTOINCREMENT,
kaf_name CHAR (40) UNIQUE ON CONFLICT IGNORE
);
```

The job table which contains possesions of the authors:

```
CREATE TABLE job (
id INTEGER PRIMARY KEY AUTOINCREMENT,
job_name CHAR (30) UNIQUE ON CONFLICT IGNORE
);
```

The table about author’s degree:

```
CREATE TABLE sci_dgr (
id INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
dgr_name CHAR (20) UNIQUE ON CONFLICT IGNORE
);
```

The graduation table which contains information about defencing:

```
CREATE TABLE graduation (
id INTEGER PRIMARY KEY AUTOINCREMENT,
stud_name CHAR (50),
supervisor_name CHAR (50),
grad_date DATE
);
```

Here the themes and the supervisor are considered to be the main tables. And other tables are logically connected to these two tables. For example, the departments linked to the authors are chosen from the department table, and the direction linked to themes is taken from category table. Additional tables are filled with lettering forecast of future.

The totally formation of tables is shown in the following tables:

Table 3.1 – Themes table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	name_uz	VARCHAR	Uzbek version of the theme
3	name_kk	VARCHAR	Karakalpak version of the theme
4	name_ru	VARCHAR	Russian version of the theme
5	name_en	VARCHAR	English version of the theme
6	Category	INTEGER	The direction that belongs to the theme
7	Supervisor	INTEGER	Author
8	Date_add	DATE	The date of being added

9	Tags	TEXT	Key words
10	Summary	TEXT	Short annotation
11	protect_info	INTEGER	The sign about defence (if it has not been carried then the sum is equal to NULL, vice versa there will be written ID in the graduation table)

Table 3.2 – supervisor table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	Name	CHAR(40)	N.S.P. of the author
3	Department	INTEGER	The name of the department
4	Job	INTEGER	Author’s position
5	Sci_dgr	INTEGER	Author’s scientific degree

Table 3.3 – category table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	Cat_name	VARCHAR	Direction

Table 3.4 – department table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	Kaf_name	CHAR(40)	The name of the department

Table 3.5 – job table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator

2	Job_name	CHAR(30)	Position

Table 3.6 – sci_dgr table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	Dgr_name	CHAR(20)	Scientific degree

Tasble 3.7 – Graduation table

№	Place name	Type	Definition
1	Id	INTEGER	Unique identificator
2	Stud_name	CHAR(50)	N.S.P. of carried student
3	Supervisor_name	CHAR(50)	Scientific supervisor
4	Grad_date	DATE	The date of defence

3.3 Organizing the interface of the user

The most comfortable method of using the database is by the interface. There are several of forms of interfaces; they are web-attachment, windowsformapplication, mobile attachments and others. Because of the fact that one user from one place is supposed to use our planned database, we needn't the web-attachment or mobile-attachment forms. So windowsformapplication form has been chosen for the interface. WE have various variants for fulfilling that. For example, Delphi, Java, MSVisualStudio other producing spheres can be used. Beause of choosing database of SQLite and being comfortable interface we choose windowsformapplication in MSVisualStudio C# language. It consists of several forms. There is main button form in database. Following points is found in this form:

- Tasks,
- Authors,
- Reports,

- Security,
- Settings,
- About program.

In Tasks we can work with all tasks in database: To look, to add, to look for, to correct, to delete and ect.

In Authors. We can save informations about athors and we can look, add, delete, correct them.

In Reports in be formed some kind of reports.

Security. We can add procted tasks in database and comment them.

Settings. We do some works with informations which is added before. For example about pulpit, direction, job title and ect.

There are informations about athor in *About program*.

Afore-mentioned every point will be made separate form. Users should pay attention to visual view of interface so that it will be comfortable. Because of that picture are added to form. To make interface we use BellMT and HighToweText fonts to be comfortable to reed in forms. The view of form looks like button stile of Windows.

3.4 Development of the forms, menu and reports

To develop interface of database is chosen MicrosoftVisualStudio 2015 and C#. Before develop program was made WindowsFormApplication and was named ProblemBase

Joining SQLite database to project

To join SQLite database with program which is done in MS Visual Studio we must download SQLite library. We download it from sqlite.org site. There are several kinds of SQLite in site:

- Manage Only – for Windows or Linux-based program,
- Compact Framework - for .NET CF programs,
- Itanium – for Itanium professors,
- x64 – x64 for platform in architecture,

- x86 – x86 for platform in architecture.

We use ManageOnly version here. Then we must join SQLite database to project. To do that we move SQLite library to referensces. This library is called System.Data.Sqlite.dll (Figure 3-6).

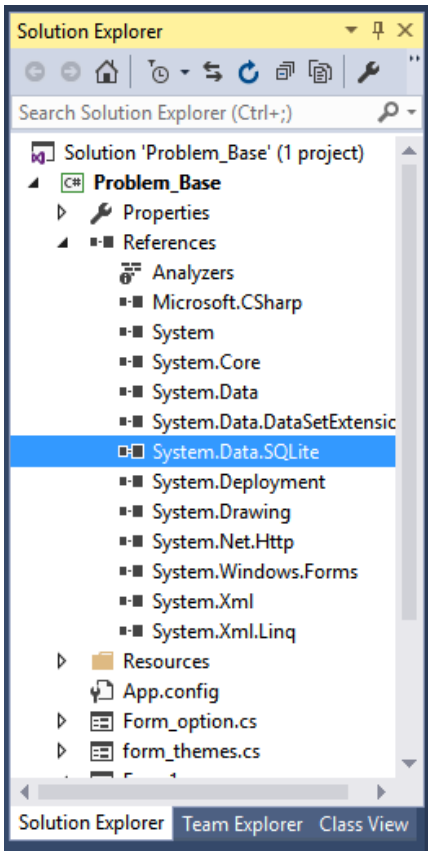

Figure 3-6. Sqlite library in References

After moving library to use it we move SQLitenamespace to project:

```
using System.Data.SQLite;
```

To connect with database we write following code:

```
publicstaticSQLiteConnection conn;
publicstaticSQLiteCommand command;
publicstaticbool db_con = false;

privatevoid Form1_Load(object sender, EventArgs e)
{
     if (File.Exists("MyDb.sqlite"))
     {
          try
          {
               conn = newSQLiteConnection("Data
               Source=MyDB.sqlite;Version=3;");

               db_con = true;
```

```
                lab_con.Text = "Подключен";
            }
            catch (SQLiteException ex)
            {
                db_con = false;
                lab_con.Text = "Не подключен";
            }

        }
        elseMessageBox.Show("Ошибка: Файл базы данных не
существует!");
}
```

There «conn» is connecting object with database. Then we can work with database by means of this object. «command» is used to implement database questions. «db_con» is logical variable and is used to define joining condition. The main window of program will be started «Form1_Load» database will be connected with program. If database exists the connecting will implement. Otherwise, it will be reported error(File.Exists()). In «conn» object will be showed name and version of database.

```
conn = newSQLiteConnection("Data Source = MyDB.sqlite;
Version=3;");
```

If the connecting is succesfull it will be shown appropriate report, otherwise it will be shown report about joining isn’t succesfull.

Making forms

For planning interface, for main form and related point will create separate buttons on higher paragraph. The picture is added seperated to database of the main form of interface. The main interface looks like this Figure 3-7.

Figure 3-7. The main window

The location of all of informations, their measurements are given on the main window. We don’t have to change them. Because them fixig stile was determined for the main window:

```
this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedSingle;
```

The started the main window of program is standed in front of the screen.

```
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
```

About connecting informations in database is shown on the left corner of the main window. If program is connected to database it will be reported «Подключен» otherwise it will be reported «Отключен» (Figure 3-8) By means of this users can control joining to database.

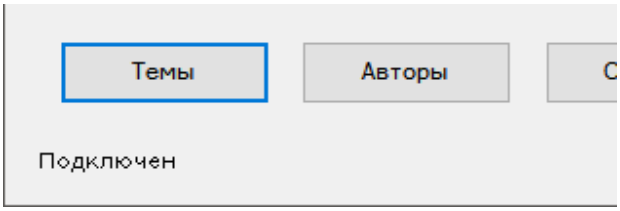

Figure 3-8. Information about connection to database

On this window we made separate buttons to following points.

- «Темы»
- «Авторы»,
- «Отчеты»,
- «Защита»,
- «Настройки»,
- «О программе».

To part of «Темы» we made apartly form. Dutys of this point are:

- To see topics,
- To add topics,
- To comment topics,
- To delete topics,
- To search topics.

This form is connected with «themes» table of database and we use in this form textBox, ComboBox, DataGridView, DateTimePicker and buttons(Figure 3-9).

To see list of informations inserted to database are used DataGridView component. This component is presented informations in the manner of table. Our informations in database are in the manner of table. The looking of DataGridView componennt is shown in Figure 3-10.

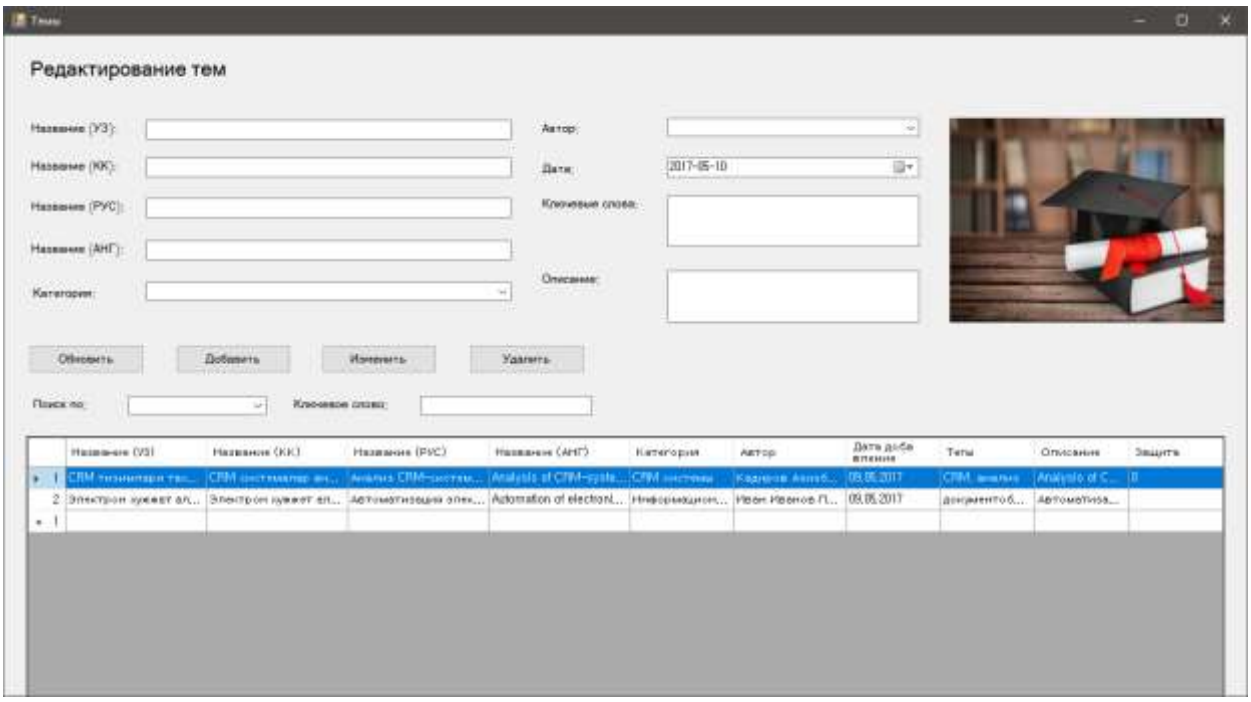

Figure 3-9. Topic editing window

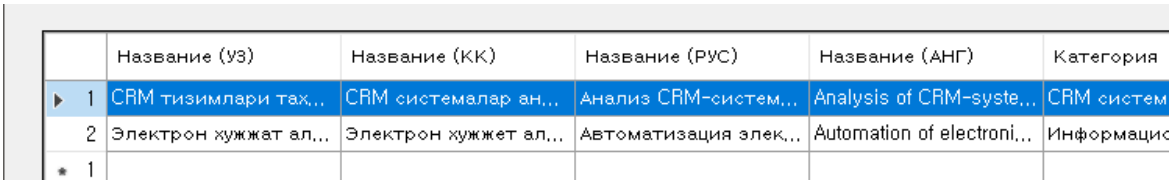

Figure 3-10. Datagridview

To present «themes» table component in database «table_show()» was made.

```
privatevoid table_show()
{
     try
     {
          Form1.conn.Open();
          DataTable dTable = newDataTable();
          String sqlQuery;
          sqlQuery = "SELECT themes.name_uz, themes.name_kk,
     themes.name_ru, themes.name_en, category.cat_name,
     supervisor.name, themes.date_add, themes.tags,
     themes.summary, themes.protect_info FROM themes, supervisor,
     category WHERE themes.category=category.id AND
     themes.supervisor=supervisor.id ORDER BY themes.name_uz";
          SQLiteDataAdapter adapter =
     newSQLiteDataAdapter(sqlQuery, Form1.conn);
          adapter.Fill(dTable);
          if (dTable.Rows.Count > 0)
          {
               dgvViewer.Rows.Clear();
               for (int i = 0; i < dTable.Rows.Count; i++)
```

```
                {
                     dgvViewer.Rows.Add(dTable.Rows[i].ItemArray);
                }
                dgvViewer.CurrentCell =
                dgvViewer.Rows[index].Cells[0];
                 }
           else
           MessageBox.Show("Database is empty");
           Form1.conn.Close();
           }
      catch (Exception ex)
      {
      MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
      MessageBoxIcon.Error);
      }
}
```

There added try and catch methods if errors happens in database not to finish the program. Form1.conn.Open() – It is for connecting with database. conn it is connecting with object of database.

DataTable this table is class is saved shown informations.

«themes» tables several grounds connected with other tables, so that crossed question was written:

```
SELECT themes.name_uz, themes.name_kk, themes.name_ru,
themes.name_en,   category.cat_name,   supervisor.name,
themes.date_add,      themes.tags,      themes.summary,
themes.protect_info FROM themes, supervisor, category
WHERE          themes.category=category.id          AND
themes.supervisor=supervisor.id ORDER BY themes.name_uz
```

SQLiteDataAdapter is class which saves in itself impotances when we send question to database. It will be made adapter object of this class. And sqlQuery question was made in database:

```
SQLiteDataAdapter adapter =
newSQLiteDataAdapter(sqlQuery, Form1.conn);
```

The next step we passed informations of dTable to DataGridView component. dgvViewer is a object of DataGridView component. For this we use cycle. Cycle is

operated for each DataGridView component and write informations of table on number column. After finishing cycle it move cursor to planned place of table:

```
dgvViewer.CurrentCell = dgvViewer.Rows[index].Cells[0];
```

CurrentCell marks the curcor there. «index» is global alternation. If new base is added value of «index» will be changed to the last line. It helps to users to see inserted informations. After writing informations in DataGridView component base and session will finished:

```
Form1.conn.Close();
```

Inserted ground in form will realizated for all grounds of «themes» table in database. «Textbox» is choosen as insert ground. List of grounds following:

- «Название (УЗ)» - TextBox component
- «Название (КК)» - TextBox component
- «Название (РУС)» - TextBox component
- «Название (АНГ)» - TextBox component
- «Категория» - ComboBox component
- «Автор» - ComboBox component
- «Дата» - DateTime component
- «Ключевые слова» - TextBox.Multiline component
- «Описание» - TextBox.Multiline component

TextBox component, which devided for topics, is installed on one number of text(Single).

AutoCompleteMode = SuggestAppend;

AutoCompleteSource=ListItems;

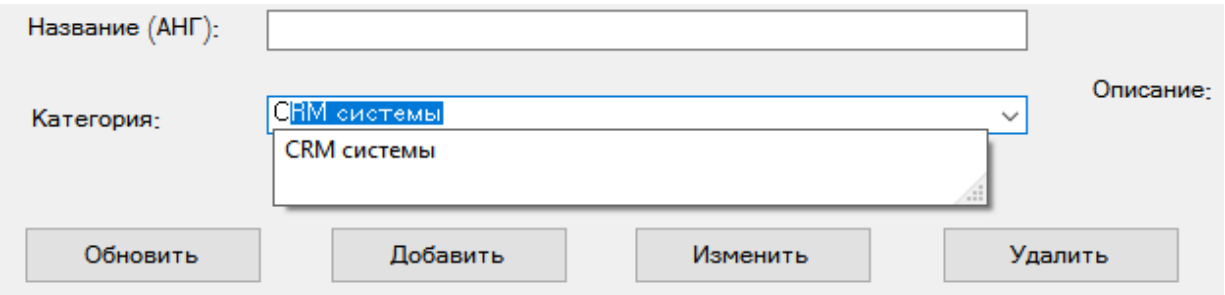

Figure 3-11. Automatically ending the topic

To fill comboBox components tables list we use «combo_show» function. We already can enter it to update it.

```
privatevoid combo_show()
        {
Form1.conn.Open();
DataTable dTable = newDataTable();
String sqlQuery;
            sqlQuery = "SELECT id, cat_name FROM category ORDER
BY cat_name";
SQLiteDataAdapter adapter = newSQLiteDataAdapter(sqlQuery,
Form1.conn);
            adapter.Fill(dTable);
            comboBox1.DataSource = dTable;
            comboBox1.DisplayMember = "cat_name";
            comboBox1.ValueMember = "id";
            comboBox1.SelectedIndex = -1;

DataTable dTable2 = newDataTable();
            sqlQuery = "SELECT id, name FROM supervisor ORDER BY
name";
SQLiteDataAdapter adapter2 = newSQLiteDataAdapter(sqlQuery,
Form1.conn);
            adapter2.Fill(dTable2);
            comboBox2.DataSource = dTable2;
            comboBox2.DisplayMember = "name";
            comboBox2.ValueMember = "id";
            comboBox2.SelectedIndex = -1;

Form1.conn.Close();
        }
```

There are two quetions, they are devided to topic directions and authors. To fill list before grounds of this list will be asked. The results of dTable and dTable2 are written here. Then comboBox component would be source of list and it will be shown:

```
comboBox1.DataSource = dTable;
```

Informations in the list can consist of two kinds: writing and its id number. For its writing was defined «cat_name» in the writing ground and its id number of ground was defined too:

```
comboBox1.DisplayMember = "cat_name";
comboBox1.ValueMember = "id";
```

We choose DateTimePicker component to know the date of insirting topic. The value of component will show users the date of computer(Figure 3-12).

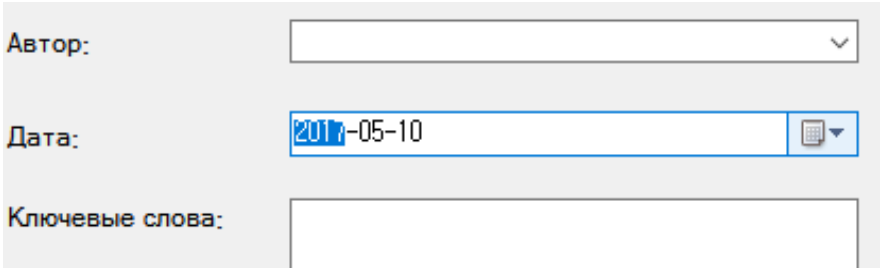

Figure 3-12. Using DateTimePicker component

When the form is working inserting grounds will be empty. If we press one of the lines of dgvViewer just this line informations will pass to insirt ground.:

```
string name_uz =
dgvViewer.Rows[e.RowIndex].Cells[0].Value.ToString();
string name_kk =
dgvViewer.Rows[e.RowIndex].Cells[1].Value.ToString();
string name_ru =
dgvViewer.Rows[e.RowIndex].Cells[2].Value.ToString();
string name_en =
dgvViewer.Rows[e.RowIndex].Cells[3].Value.ToString();
string category =
dgvViewer.Rows[e.RowIndex].Cells[4].Value.ToString();
string author =
dgvViewer.Rows[e.RowIndex].Cells[5].Value.ToString();
string date =
dgvViewer.Rows[e.RowIndex].Cells[6].Value.ToString();
string tag =
dgvViewer.Rows[e.RowIndex].Cells[7].Value.ToString();
string summary =
dgvViewer.Rows[e.RowIndex].Cells[8].Value.ToString();
                    combo_show();
                    comboBox1.SelectedText = category;
                    comboBox2.SelectedText = author;


                    textBox1.Text = name_uz;
                    textBox2.Text = name_kk;
                    textBox3.Text = name_ru;
                    textBox4.Text = name_en;
                    textBox5.Text = tag;
textBox6.Text = summary;
```

And then we can work with this informations.

We make four buttons in the form, they are:

- «Обновить» - to load the writing in table;

- «Добавить» - to add new topic to base;
- «Изменить» - to correct informations;
- «Удалить» - to delete topic.

When we press *«Добавить»* button informations in insirt ground will write in database.

```
string sql = "INSERT INTO themes('name_uz', 'name_kk',
'name_ru', 'name_en', 'category', 'supervisor', 'date_add',
'tags', 'summary') VALUES ('" + name_uz + "','" + name_kk +
"','" + name_ru + "','" + name_en + "', (SELECT id FROM category
WHERE cat_name = '" + category + "'),(SELECT id FROM supervisor
WHERE name = '" + author+ "'),'" + this.dateTimePicker1.Text +
"','"+ tag +"','"+ summary +"')";
```

Several grounds are connected with other tables.In table «themes» will save id number of other tables. From information of insirt ground we can define id number and we must write just this id number in database. After writing question we write following code to work with it:

```
Form1.command = newSQLiteCommand(sql, Form1.conn);
Form1.command.ExecuteNonQuery();
Form1.conn.Close();
index = dgvViewer.RowCount - 1;
table_show();
```

There will be made query(Form1.command.ExecuteNonQuery();) and session qill ended with base.

After pressing «Изменить» button informations in database will updated with informations in insirt ground. To change and choose the topic we must press one of the lines in dgvViewer table. When choosing line this order number will write «index» change.And because of this number uzbek name of topic will temporary write:

```
string cur_index =
dgvViewer.Rows[index].Cells[0].Value.ToString();
```

Because of this name of topic is unical. Then gounds in database will be updated:

```
string sql = "UPDATE themes SET name_uz='" + name_uz + "',
name_kk='"+name_kk+"',name_ru='"+name_ru+"',name_en='"+name_en+"
```

```
',category=(SELECT id FROM category WHERE cat_name='" + category
+ "'), supervisor=(SELECT id FROM supervisor WHERE name='" +
author + "'), date_add='"+date+"', tags='"+tag+"',
summary='"+summary+"' WHERE name_uz='" + cur_index + "'";
```

When we press «Удалить» button grounds are deleted review on uzbek name:

```
string name_uz = textBox1.Text;

string sql = "DELETE FROM themes where name_uz='" + name_uz +
"'";
```

To search for

To look for informations in database we use two grounds in this form, textBox and comboBox components. We use textBox to insirt searching informations, comboBox to know what ground is concerned searching information. We shouldn't insirt searching topic in graund or press some buttons. To do that we insirt just topic and immediately look on having informations in this table. If topic of table changes(textBox_TextChanged) we will write code necessary for working metod.

```
Form1.conn.Open();
DataTable dTable = newDataTable();
String sqlQuery;


if (comboBox3.Text == "Название (УЗ)")
                {

                    sqlQuery = "SELECT themes.name_uz,
themes.name_kk, themes.name_ru, themes.name_en,
category.cat_name, supervisor.name, themes.date_add,
themes.tags, themes.summary, themes.protect_info FROM themes,
supervisor, category WHERE themes.category = category.id AND
themes.supervisor = supervisor.id AND themes.name_uz LIKE '" +
"%" + textBox7.Text + "%" + "' ORDER BY themes.name_uz";
SQLiteDataAdapter adapter = newSQLiteDataAdapter(sqlQuery,
Form1.conn);
adapter.Fill(dTable);
                }

dgvViewer.Rows.Clear();
for (int i = 0; i < dTable.Rows.Count; i++)
                {
```

```
dgvViewer.Rows.Add(dTable.Rows[i].ItemArray);
                }
                dgvViewer.CurrentCell =
dgvViewer.Rows[index].Cells[0];

Form1.conn.Close();
```

Firstly we check choosing writing of comboBox . For example, if the name is chosen in Uzbek language of the table we will search for just about this graund. Searching will proccesed in question form. We use LIKE order to search, because it is comportable. For example, user remembered just one topics name and by means of this he must find all name.

```
LIKE '" + "%" + textBox7.Text + "%"
```

This «%» symbol means free symbols. The result of searching we can see on the table (Figure 3-13).

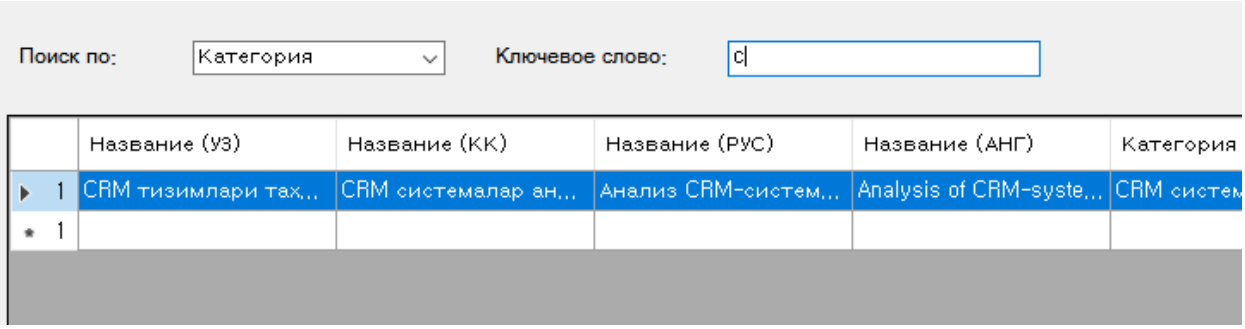

Figure 3-13. Search by keyword

CHAPTER 4. OCCUPATIONAL SAFETY AND HEALTH

4.1 Occupational safety

Occupational safety is a system of legislative acts, socio-economic, organizational, technical, hygienic, and therapeutic and preventive measures and tools that ensure the safety, health and human performance in the work process.

In practical activities, it is customary to consider legislative acts and socio-economic measures in terms of labor legislation and management of labor protection; Technical measures and means - safety engineering and fire protection system; Hygienic and therapeutic and preventive measures and means - occupational health and industrial sanitation: organizational measures - in terms of both safety and industrial sanitation practices and occupational safety management.

All the variety of legislative acts, measures and means included in the concept of labor protection, is aimed at creating such working conditions, in which the impact on working dangerous and harmful production factors is excluded.

Labor legislation is a set of existing laws and regulations that regulate the social and labor relations of workers and employees.

The labor legislation regulates the labor relations of all workers and employees, promoting the growth of labor productivity, raising production efficiency and raising the material and cultural standard of living of workers on this basis, and strengthening labor discipline. It establishes a temporary protection of the right to work of workers and employees and an inalienable right to create a high level of safety and healthy working conditions for them.

The right of citizens to work is realized through the conclusion by workers and employees of a labor contract on a robot at the relevant enterprise, institution and organization. The labor wars agreement is an agreement between workers and an enterprise, according to which a worker or employee undertakes to perform work in a certain specialty, qualification or position, subject to internal regulations.

The enterprise under this agreement undertakes to pay the worker wages and provide the conditions provided for by the labor legislation with the collective agreement and agreement of the parties.

In accordance with the labor legislation, healthy and safe working conditions should be created at all enterprises, institutions and organizations. The provision of such conditions is entrusted to the administration, which is obliged to introduce modern

Safety precautions, preventive injuries and create sanitary and hygienic conditions that prevent the occurrence of occupational diseases.

The most important condition for ensuring labor safety is training, instruction and testing of knowledge on labor protection. Persons recruited or transferred to another, pass the following types, briefing: introductory, primary at the workplace, repeated, unscheduled and current. Officials guilty of violating labor laws, regulations on safety and industrial sanitation, failing to fulfill their obligations under collective agreements and agreements on labor protection or in obstructing the activities of trade unions are responsible for this. Depending on the degree of guilt, they can be brought to disciplinary, administrative, material and criminal liability.

Requirements for the workplace of the programmer.

Tables should be quite large - from 1200x800 mm to 1600x800 mm. On this table you can place all your equipment. It is on the table, and not under the table, as we usually do, because under the table should be quite spacious legs and even climb under the table to insert a floppy disk into the drive, is also very tiring. It is now fashionable to have a table under the table with a slope of 5 to 15 degrees for the feet, which provides comfort and peace to the feet during a long sitting at the computer. The chair should be mounted on wheels, the height of the seat from the floor is 42-53 cm, the height of the table is 72 cm. To sit down for a computer it is necessary so that the angle between a trunk and hips, a trunk and a forearm was equaled to 90 degrees. The monitor should be at least 45cm away from the eyes, and it is advisable to tilt it so that you are not looking at the monitor with your eyes

facing the eye, but looking at it from above at an angle of 5 to 35 degrees to the screen surface. The angle of the keyboard to the table is recommended to be set within 5-10 degrees.

Safety is a system of organizational measures and technical means, prevention of exposure to work in hazardous work areas or factors.

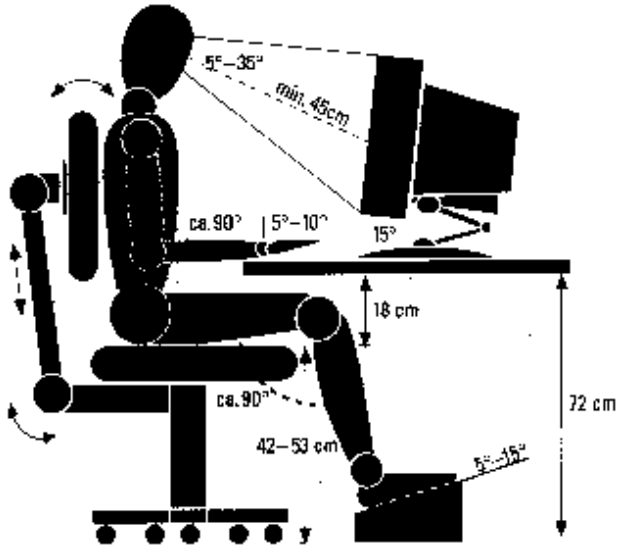

Figure 4-1. Body position when working at the computer

There are 5 types of safety briefings:

- Introductory briefing (for all, regardless of education or length of service);
- Primary - in the workplace (on the day of employment);
- Repeated (at least once every 6 months);
- Unplanned (when changing the rules of labor protection);
- Current (spend with employees before the production of work, which formalizes the order-admission).

When working on a PC in a room, the temperature rises and decreases. Relative air humidity, ionic and qualitative composition of air deteriorates; The content of organic substances and carbon dioxide in the air increases. The content in the air of these substances can exceed the permissible limit by several times. For

this reason it is necessary to maintain the following optimum microclimate parameters: air temperature from 18 to 21 ° C; Relative humidity of air, respectively, 55-62%; The air speed is less than 0.1 m / s. Also, ventilation of the room should be carried out, depending on the weather conditions,

Duration must be at least 10 minutes. The best air exchange is carried out through through ventilation.

Another way to ensure air exchange, can be achieved by installing in window openings of independent air conditioners.

Sound insulation of the enclosing structures of the ITC must also meet certain requirements. To reduce the noise level, the ceiling or walls above 1.5 - 1.7 meters from the floor should be lined with sound absorbing material with the maximum coefficient

Sound absorption in the frequency range 63-8000 Hz. Additional sound absorption in the ITC can be curtains suspended in the fold at a distance of 15-20 cm from the fence, made of dense, heavy fabric. For a reduced absorption of light, the ceiling and walls above the panels (3,5 - 1.7 m.), If they are not lined with sound-absorbing material, are painted with white water-based paint (the reflection coefficient should be at least 0.7). For painting the wall panels it is recommended to give preference to light colors.

In lighting installations (OC) ITC should use a system of general lighting, made by ceiling or suspended fluorescent lights, evenly placed on the ceiling in rows parallel to the light holes so that the screen of the video monitor was in the zone of the protective angle of the luminaire and its projection did not occur on the screen.

Working on a PC should not see the reflection of the fixtures on the screen. Apply local lighting when working on a PC is not recommended.

Work on PC can be carried out with the following types of lighting:

- General fluorescent lighting, when video monitors are located around the perimeter of the room or at the central location of workplaces in two rows along the length of the class with screens facing in opposite directions;

- Combined lighting (natural + artificial) only with one and three rows of workplaces, when the screen and the surface of the working table are perpendicular to the light-bearing wall.

Natural lighting, when workplaces with PC are located in a row along the length of the room at a distance of 0.8 - 1.0 m from the wall with window openings, and the screens are perpendicular to this wall. The main stream of natural light at this should be on the left. Do not direct the main light stream of natural light on the right, behind and in front working on the PC. The optimal distance of the eyes to the monitor screen should be 60-70 cm, permissible not less than 50 cm. It is not recommended to consider information closer than 50 cm.

Recommendations on the use of computer technology:

- It is necessary to observe restrictions on work with personal computers for employees suffering from diseases of the musculoskeletal system, eyes, skin, and also for pregnant women;
- it is preferable to use displays with high resolution and a screen size of at least 14 "(Hi-Resolution, Non-Interlaced);
- it is better to choose video adapters with high resolution and frame refresh rate not less than 70 ~ 72Hz;
- it is obligatory to place screen filters with antistatic coating on the displays, reducing eye fatigue and concentration of dust particles near the monitor screen several times;
- sit no closer than 70 cm from the display:
- the display screen should be oriented in such a way as to exclude glare from light sources;
- do not place the display directly under the light source or close to it;
- it is desirable that the illumination of the operator's workplace does not exceed 2/3 of the normal illumination of the room;
- The wall behind the display should be lit about the same as its screen.

- when placing several personal computers in the same room, the distance from the workplace of each operator to the back and side walls of neighboring personal computers must be at least 1.2 m;
- the workplace should be equipped so as to exclude uncomfortable postures and prolonged static body stresses;
- the total working time with the display should not exceed 50% of the operator's total working time;
- Do not exceed the work rate of about 10 thousand keystrokes and an hour (about 1500 words);
- with the usual work with a computer, you need to do 15-minute breaks every 2 hours, and with intensive work - every hour.

In addition to the security of the user, it is necessary to say a few words about the security of the computer and, most importantly, the security of the data stored in it. PCs used to store critical information need to be equipped with uninterruptible power supplies that support the supply voltage for some time in case of emergency in the electrical network. You can not block the back of the system unit or put a personal computer close to the wall -

This leads to a "heavy" mode of cooling the system unit and its overheating. The same goes for the display - you can not put paper, books, or anything else that can block its ventilation holes. Dust and electronics are poorly compatible with each other, therefore it is necessary to maintain an acceptable dust regime in the room.

When there is a smell of burning, immediately stop work, turn off the equipment.

Before you start, make sure there are no visible damage to the workstation, sit down so that the line of sight is at the center of the screen, so that you do not bend down to use the keyboard and perceive the information sent to the monitor.

During operation, follow the above rules, monitor the operation of the equipment and immediately stop working when there is an outside sound or spontaneous shutdown of the equipment.

Illumination of ITC is also of great importance when working on a PC. It is largely determined by the color and network environment. For a reduced absorption of light, the ceiling and walls above the panels (1.5-1.7 m), if they are not lined with sound-absorbing material, are painted with white water-based paint (the reflection coefficient should be at least 0.7). For painting the wall panels it is recommended to give preference to light colors. In lighting installations (DU) ITC should use a system of general lighting, filled with ceiling or hanging fluorescent lights, evenly placed on the ceiling in rows parallel to the light holes so that the screen of the video monitor was in the zone of the protective angle of the luminaire and its projection was not on the screen.

Working on a PC should not see the reflection of the fixtures on the screen. Apply local lighting when working on a PC is not recommended.

The main stream of natural light at this should be on the left. Do not direct the main light stream of natural light on the right, behind and in front working on the PC.

Work on PC during classes or during production practice can be carried out with the following types of lighting: common fluorescent lighting - video monitors are located around the perimeter of the room or with a central arrangement of workplaces in two rows along the length of the class with screens facing in opposite directions; Combined lighting, (natural + artificial) only with one or three rows of workplaces where the screen and the surface of the working table are perpendicular to the light-bearing wall; Natural lighting when workstations with a PC are arranged in a row along the length of the class at a distance of 0.8-1.0 m from the wall with window openings, and the screens are perpendicular to this wall. The main stream of natural light with this lighting should be on the left. The optimal distance of the eyes to the monitor screen should be 60-70 cm, permissible not less than 50 cm. It is not recommended to consider information closer than 50 cm.

4.2. Fire safety when working on a personal computer.

Fires in the computer center are particularly dangerous, since they are well-ordered with large material losses. A characteristic feature of small areas of premises. As is known, a fire can occur during the interaction of substances, oxidant, and ignition sources. The combustible components are: building materials for acoustic and aesthetic finishing of rooms, partitions, floors, doors, insulation of power, signal cables, etc. To remove heat from the computer, a powerful air conditioning system operates. Therefore, oxygen, the oxidizing agent of combustion processes, is available at any point in the premises of the VC.

The peculiarity of computers is the very high density of the location of microcircuits. When electricity flows through conductors and parts, heat is released, which, under conditions of high density, can lead to overheating. Reliable operation of individual elements and microcircuits as a whole is provided only in certain intervals of temperature, humidity and at specified electrical parameters. If the actual operating conditions deviate from the calculated ones, there may be fire dangerous situations.

Cable lines are the most dangerous place. The presence of combustible insulating material, probable sources of ignition in the form of electric sparks and arcs, branching and inaccessibility make the cable lines the place of the most likely occurrence and development of a fire. To reduce the flammability and the ability to spread the flame, the cables are covered with flame retardant coatings.

Fire is uncontrolled burning outside a special hearth, which causes great damage. The main reason for the non-electric fire is careless handling of fire, as well as explosions of gas-air and steam-air mixtures. Electrical combustion is the closure, overload of electric current on electrical equipment, thunder lightning.

Elimination of the causes of fire in electrical equipment is carried out in various directions:

Prevention of closure is made by the right choice, installation of network operation;

Application of circuit protection in the form of high-speed relays, as well as switches, fuses, circuit breakers. Important attention should be paid to the fire safety of the enterprise as a whole and its individual premises. Garbage, unnecessary papers, trash and other things that are not used in the production process should not accumulate in the premises. It is necessary to provide emergency exit from the premises in case of fire. Fire extinguishers should be provided in the room. They should be in working condition and checked according to the norms. There must be a fire alarm in the rooms. In case of fire, it is necessary to inform the nearest fire department and, if possible, take some steps to eliminate it.

Since the work of the programmer is directly related to electrical equipment, it is necessary to be able to use it correctly and to observe safety measures against the electric current damage.

There are many measures against electrical shock. One of them is protective earthing. Protective earthing - deliberately electrical grounding with earth or its equivalent of metal non-conductive parts, which may be energized due to a short circuit to the enclosure and for other reasons.

The purpose of protective earthing is to eliminate the danger of electric shock in case of contact with the body and other live parts of the electrical installation that are under voltage. The principle of protective earthing is the reduction to safe values of touching voltages and pitch, caused by the closure of the case and other reasons. This is achieved by reducing the potential of grounded equipment (by raising the potential of the base on which the person stands, to a value close to

Value of the potential of grounded equipment).

There is also such a measure of protection against electric shock, called zanuneniem. The danger of electric shock when touched to the housing and other non-current metal parts of electrical equipment that are under voltage due to the closure on the housing and for other reasons can be eliminated by the rapid disconnection of the damaged electrical installation from the mains and, at the same time, the voltage drop of the housing relative to the ground. This goal is also

zeroing. Zeroing is a deliberate electrical connection with a zero protective conductor of metal non-conductive parts, which may be energized.

A zero protective conductor is a conductor that connects the nullable parts to a dull-grounded neutral point of the current source or its equivalent.

The principle of the action of zeroing is the transformation of the closure in order to cause a large current capable of triggering the protection and thus automatically shutting down. Damaged electrical installation from the mains.

Also there is such a means of protection as safety shutdown. Protective switch-off - high-speed protection, ensuring the disconnection of the electrical installation when there is a danger of human injury by electric shock. Such a danger can arise, in particular, when the phase is closed on the case, the insulation of the network is lower than a certain limit, and, finally, in the case of a person touching the current-carrying part under tension. The main elements of the residual current device are the residual current device and the circuit breaker.

The device of protective shutdown - a set of individual elements that perceive the input value, responds to its changes and at a given value give a signal for its disconnection of the switch. These elements are:

- The sensor is the input link of the device that receives the effects from outside and carries out the transformation of this effect into the corresponding signal;
- An amplifier designed to amplify a sensor signal if it is not powerful enough;
- Control circuits serving periodic checks of the serviceability of the protective shutdown;

Auxiliary elements - signal lamps and measuring instruments characterizing the condition of the electrical installation. Automatic circuit-breaker is a device intended for switching on and off from circuits under load and in case of short circuits. It must turn on the circuit automatically upon arrival signal from the residual current device. Also there are various electrical protective devices against

electric shock. Protective equipment can be conditionally divided into three groups: insulating, enclosing and protective.

Isolating - isolate the person from current-carrying or grounded parts, as well as from the ground. They are divided into basic and additional.

The main ones are insulated, able to withstand the working voltage of the electrical installation for a long time and therefore they are allowed to touch live live parts. These include: in electrical installations up to 1000 W - dielectric gloves, insulating rods, insulating and electrical clamps, etc .; More than 1000 W - insulating rods, and electrical clamps, as well as tools for repair work at a voltage of more than 1000 W.

Extra - have insulation unable to withstand working voltage. Electrical installations, and therefore they can not independently protect a person from electric shock by this voltage. Their importance is to strengthen the protective actions of the basic and isolating means, with which they should be applied, and with the use of basic protective means, it is sufficient to use one additional protective device. Additional facilities include electrical installations up to 1000 Watt - dielectric galoshes of carpets, as well as insulating stands.

4.3. Activities conducted to protect the environment.

Environmental protection is a complex of scientific and technical, industrial and economic and administrative-legal measures aimed at the conservation and proper use of land and its subsoil, water resources, flora and fauna, ensuring the purity of air and water, the reproduction of natural resources, harmonious relationships between Society and nature. It is not difficult to imagine a prosperous economy in which people become more and more sick from year to year as a result of an incorrect approach to their health and a polluted environment. As we build our society, we need to use our growing efforts to ensure that our citizens are healthy throughout their lives and surrounded by a healthy natural environment.

The source of atmospheric air pollution is a technological unit that releases harmful substances during operation.

The amount of substances allocated depends on the type and capacity of production, its technical equipment and is determined by instrumental measurements or calculations using special industry methods. The main measures to reduce emissions of harmful substances into the atmosphere are the improvement of technological processes, including the reduction of industrial emissions; Construction of new and increasing efficiency of existing treatment facilities; Elimination of pollution sources, re-profiling of production. With respect to the enterprises of machine-building production, the most important is gas - and dust removal of ventilation emissions, especially with open casting of metals.

Purification and neutralization of gas components of emissions from industrial production is carried out by methods whose choice is determined by the composition, concentration of pollutants, type of production, emission conditions.

Purification of ventilation emissions from mechanical impurities is carried out by devices, wet and dry dust collection, fibrous filters and electrostatic precipitators.

As filters, various filtering, thin coarse fibrous materials are used. In addition, electrofilters are widely used at machine-building enterprises, which, depending on the method for removing precipitated particles on electrodes, are divided into dry and wet.

Protection of drinking water is also of global importance. The degree of wastewater treatment is set depending on local conditions, taking into account the possible use of treated wastewater for industrial and agricultural needs.

At the enterprises of the metallurgical and machine-building industry, wastewater treatment is carried out, as a rule, in sedimentation tanks, slag storage tanks, oil-oil traps using a number of cases of coagulants. The resulting slag, containing a large number of metals, is utilized and included in the composition of the charge. Purified water is in most cases used in circulating water supply

systems. In this case, the water from the main source or from other cycles of water use is used to compensate for the circulating water losses

.

An important direction of environmental protection is the protection of soils from water and wind erosion, combating their salinization through the introduction of appropriate crop rotation, the creation of forest shelter belts, fastening and afforestation of gullies and gullies, and the use of reclamation facilities.

In order to reduce soil pollution with various industrial wastes, the following measures are envisaged in the practice of protecting land resources: utilization, incineration, landfilling at special landfills.

The choice of the method for neutralizing and recycling waste depends on their chemical composition and the degree of their impact on the environment. In a number of cases, waste from machine-building and metallurgical industries contains a significant number of chemical compounds that can be of value as raw materials and used as a supplement to the charge.

CONCLUSION

The main aim of this practice work is creating database which consists of tasks of diploma works and using it in practice. To do this we work with our scientific supervisor.

Following problems will solved to create this graduating practice work:

- Examining the activities of higher education institutions qualification proceedings studied;
- Studied the process of selection of topics studied in higher education institutions;
- The structure of the problem for database is determined;
- Database management system and its interface software tool selected;
- The database and interface software was made;
- The program is tested;

Analogs, in our case, the favorable SQLite database is selected. The database interface for MS Visual Studio C# programming language developed interface program.

The first data into the database and application tested. The errors occurred in the process of testing the program. The work is presented for employees responsible for work with qualification papers in Tashkent University of Information Technologies named after Muhammad al-Khwarizmi Nukus branch and evualuated.

As a result of final qualifying paper and performance of the tasks put forward the goal of final qualifying work.

REFERENCES

1. Michael Owens. The Definitive Guideto SQLite. Apress Berkely, CA, USA 2010.
2. Sibsankar Haldar. SQLite Database System Design and Implementation. Books.google.com. USA 2015
3. Jay A. Kreibich. Using SQLite. O'Reilly Media. 2010.
4. Rick F. van der Lans. The SQL Guide to SQLite. Lulu.com. 2009
5. Alex Mackey Introducing .NET 4.0: with Visual Studio 2010; Apress - , 2010.
   Andrew Filev Professional UML Using Visual Studio .Net; - Москва, 2012.
6. Brain Johnson Working with Microsoft Visual Studio 2005; - Москва, 2010.
7. Chris Sells, Jon Flanders, Ian Griffiths Mastering Visual Studio .NET - Москва, 2010.
8. Istvan Novak Beginning Visual Studio 2010 LightSwitch Development; M.: Медгиз. - Москва, 2011.
9. Joseph Albahari, Ben Albahari. C# 6.0 in a Nutshell, 6th Edition. O'Reilly Media. 2015.
10. Jennifer Greene, Andrew Stellman. Head First C#. O'Reilly Media. 2007.
11. Stephen Cleary. Concurrency in C# Cookbook. O'Reilly Media. 2014.
12. Jesse Liberty, Brian MacDonald. Learning C# 3.0. O'Reilly Media. 2008.
13. Eugene Agafonov, Andrew Koryavchenko. Mastering C# Concurrency. Packt Publishing. 2015.
14. Benjamin Perkins, Jacob Vibe Hammer, Jon D. Reid. Beginning C# 6 Programming with Visual Studio 2015. Wiley / Wrox. 2015.
15. C.J. Date. The New Relational Database Dictionary. O'Reilly Media. 2015
16. Jan L. Harrington. Relational Database Design and Implementation, 4th Edition. Elsevier / Morgan Kaufmann. 2016.
17. Martin Holzke. SQL Database for Beginners. Packt Publishing. 2016
18. Patrick O'Neil. Database. Elsevier / Morgan Kaufmann. 2014.

19. Michael Schmalz. C# Database Basics. O'Reilly Media. 2012.
20. C.J. Date. Database Design and Relational Theory. O'Reilly Media. 2012.

21. https://www.sqlite.org/docs.html
22. https://msdn.microsoft.com