# Московский технологический институт

# Базы данных № 3-4
# «Язык SQL»

**Автор:** к.т.н., доцент Долин Георгий Аркадьевич

**Контакты:** dolin1974@gmail.com

# Литература


- Королева О.Н. Базы данных [Электронный ресурс]: курс лекций/ Королева О. Н., Мажукин А.В., Королева Т.В.— Электрон. текстовые данные.— М.: Московский гуманитарный университет, 2012.— 66 с.
- Основы современных баз данных [Электронный ресурс]: методическая разработка к выполнению лабораторных работ (№1-3)/ — Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013.— 37 с.
- Темирова Л.Г. Базы данных [Электронный ресурс]: учебно-методическое пособие для выполнения лабораторных работ для студентов III курса обучающихся по направлению подготовки 231300.62 Прикладная математика/ Темирова Л.Г.— Электрон. текстовые данные.— Черкесск: Северо-Кавказская государственная гуманитарно-технологическая академия, 2014.— 57 с.
- Швецов В.И. Базы данных [Электронный ресурс]/ Швецов В.И.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 218 с.

- Выполнение запросов
- Язык SQL
- Проектирование, разработка и обслуживание баз данных

# Основные преимущества SQL

- Его поддерживают многие СУБД
- Не зависит от физического размещение данных
- Реляционная основа языка
- Дает возможность динамически менять и расширять базу данных
- Поддерживает архитектуру клиент-сервер.

# Пять основных частей SQL

- DDL – язык определения данных. Позволяет создавать, изменять, удалять объекты: таблицы, связи между таблицами. Операторы: CREATE, ALTER, DROP.
- DML – язык манипулирования данными. Позволяет добавлять, изменять, удалять записи в таблицах: INSERT, DELETE, UPDATE
- DQL – язык запросов. Позволяет получать данные из таблиц с помощью оператора SELECT.
- DCL – язык управления доступом. GRANT и REVOKE
- Transaction Control - язык управления транзакциями: COMMIT, ROLLBACK.

## Язык SQL

SQL (англ. Structured [English] Query Language — «[английский] язык структурированных запросов») — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. SQL основывается на теории множеств, но не является реляционным.

Разработан в начале 70-х. Первый стандарт SQL-86.

SQL – информационно-логический язык.

Преимущества:

- независимость от конкретной СУБД;
- наличие стандартов (SQL:2003 Core – 1300 стр.);
- декларативность – «что», а не «как».

# SQL – Structured Query Language

- SQL – это структурированный язык запросов к реляционным базам данных (БД).
- SQL – декларативный язык, основанный на операциях реляционной алгебры.
- Стандарты SQL, определённые Американским национальным институтом стандартов (ANSI):
  - SQL-1 (SQL/89) – первый вариант стандарта.
  - SQL-2 (SQL/92) – основной расширенный стандарт.
  - SQL-3 (SQL/1999, SQL/2003) – относится к объектно-реляционной модели данных.
- Подмножества языка SQL:
  - DDL (Data Definition Language) – команды создания/изменения/удаления объектов базы данных (create/alter/drop);
  - DML (Data Manipulation Language) – команды добавления/модификации/удаления данных (insert/update/delete), а также команда извлечения данных select;
  - DCL (Data Control Language) – команды управления данными (установка/снятие ограничений целостности). Входит в подмножество DDL.

# Работа с SQL

- Особенности синтаксиса:
  - В командах SQL не различаются прописные и строчные буквы (кроме содержимого символьных строк).
  - Каждая команда может занимать несколько строк и заканчивается символом ';'.
  - Символ и символьная строка заключается в одинарные кавычки:
  - 'A', '2' , 'строка', 'другая строка'
  - Однострочный комментарий начинается с символов '--'.
  - Многострочный комментарий заключается в символы /* ... */.

# Команды DDL

- CREATE – создание объекта.
- ALTER – изменения структуры объекта.
- DROP – удаление объекта.
- Общий вид синтаксиса команд DDL:

- create
- alter тип_объекта имя_объекта [параметры];
- drop

# Создание таблиц

- CREATE TABLE [имя_схемы.]имя_таблицы
- ( имя_поля тип_данных [(размер)] [NOT NULL]
- [DEFAULT выражение]
- [ограничения_целостности_поля…]
- .,..
- [, ограничения_целостности_таблицы .,..]
- )
- [ параметры ];

- ограничения_целостности (ОЦ):
- [CONSTRAINT имя_ОЦ ] название_ОЦ [параметры]

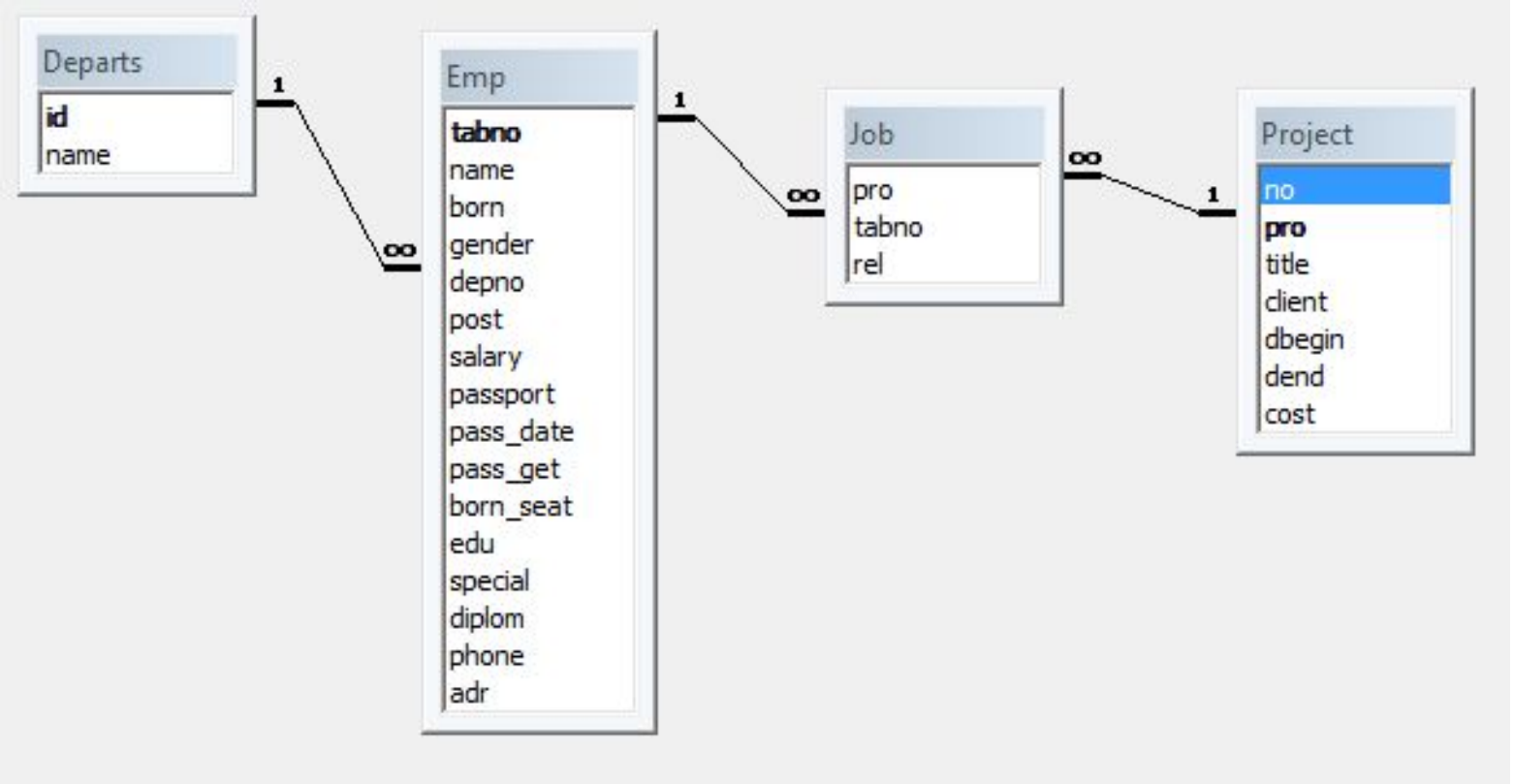
# Типы данных

- Символьные типы:
  - CHAR [(длина)] – строка фиксированной длины.
  - Длина по умолчанию – 1, максимальная длина 2000 б.
  - Строка дописывается до указанной длины пробелами.
  - VARCHAR2 (длина) – строка переменной длины.
  - Максимальная длина 4000 б. Хранятся только значащие символы.
- Числовой тип:
  - NUMBER [(точность[, масштаб])] – используется для представления
  - чисел с заданной точностью.
  - Точность по умолчанию 38, масштаб по умолчанию – 0.
  - number(4) – числа от -999 до 9999
  - number(8,2) – числа от -99999.99 до 999999.99
- DATE – дата и время с точностью до секунды. Занимает 7 байт.
  - sysdate – функция получения текущих даты и времени.
  - Тип date поддерживает арифметику дат:
  - sysdate+1 – завтра
  - (дата1 – дата2) – количество дней, прошедших между двумя датами
  - (sysdate – 0.5) – 12 часов назад

# Ограничения целостности

- В СУБД Oracle поддерживаются следующие ограничения целостности:
  - уникальность (значений атрибута или комбинации значений атрибутов):
  - UNIQUE (имя_атрибута1 [, имя_атрибута2,...])
  - обязательность / необязательность:
  - NOT NULL / NULL
  - первичный ключ:
  - PRIMARY KEY(имя_атрибута1 [, имя_атрибута2,...])
  - внешний ключ:
  - FOREIGN KEY(имя_атрибута1 [, имя_атрибута2,...]) REFERENCES имя_таблицы [(имя_атрибута1 [, имя_атрибута2,...])]
  - условие на значение поля:
  - CHECK (условие)
  - Например: check (salary>=4500), check (date2 > date1)

# Пример БД: проектная организация




Departs – отделы,

Emp – сотрудники,

Project – проекты,

Job – участие в проектах.

# Пример БД: проектная организация

**Emp** – сотрудники:

**tabno** – табельный номер сотрудника, первичный ключ;
**name** – ФИО сотрудника, обязательное поле;
**born** – дата рождения сотрудника, обязательное поле;
**gender** – пол сотрудника, обязательное поле;
**depno** – номер отдела, обязательное поле, внешний ключ;
**post** – должность сотрудника;
**salary** – оклад, больше МРОТ;
**passport** – серия и номер паспорта, уникальный обязательный атрибут;
**pass_date** – дата выдачи паспорта, обязательное поле;
**pass_get** – кем выдан паспорт, обязательное поле;
**born_seat** – место рождения сотрудника;
**edu** – образование сотрудника;
**special** – специальность по образованию;
**diplom** – номер диплома;
**phone** – телефоны сотрудника;
**adr** – адрес сотрудника;
**edate** – дата вступления в должность, обязательное поле.

# Пример БД: проектная организация

**Departs** – отделы:

**did** – номер отдела, первичный ключ;
**name** – название отдела, обязательное поле.

**Project** – проекты:

**No** – номер проекта, первичный ключ;
**title** – название проекта, обязательное поле;
**pro** – краткое название проекта, обязательное уникальное поле;
**client** – заказчик, обязательное поле;
**dbegin** – дата начала выполнения проекта, обязательное поле;
**dend** – дата завершения проекта, обязательное поле;
**cost** – стоимость проекта, обязательное поле.

**Job** – участие в проектах:

**pro** – краткое название проекта, внешний ключ;
**tabNo** – номер сотрудника, участвующего в проекте, внешний ключ;
**rel** – роль сотрудника в проекте; может принимать одно из трех значений:
'исполнитель', 'руководитель', 'консультант'.
Первичный ключ – комбинация полей **pro** и **tabNo**.

# Создание таблиц БД проектной организации

- Таблица «Отделы» (Depart):
- create table depart (did number(4) constraint pk_depart  PRIMARY KEY,
- name varchar2(100) not null
- );
- Таблица «Сотрудники» (Emp):
- create table emp ( tabno number(6) constraint pk_emp PRIMARY KEY,
- name varchar2(100) not null,
- born date not null,
- gender char  not null,
- depno number(4) not null constraint fk_depart  REFERENCES depart,
- post varchar(50) not null,
- salary number(8,2) not null constraint check_sal  check (salary > 4630),
- passport char(10) not null constraint passp_uniq  UNIQUE,
- pass_date date not null,  pass_get varchar2(100) not null,
- born_seat varchar2(100),edu varchar2(30),
- special varchar2(100),  diplom varchar2(40),
- phone varchar2(30),  adr varchar2(80),
- edate date  not null  default  trunc(sysdate),
- chief  number(6)  constraint  fk_emp  REFERENCES  emp
- );

# Создание таблиц БД проектной организации

- Таблица «Проекты» (Project):
- create table project (No number(5) constraint pk_project  primary key,
- title varchar2(200) not null,
- pro varchar(15) not null constraint  pro_uniq unique,
- client varchar(100) not null,
- dbegin date not null,
- dend date not null,
- cost number(9)
- );
- Таблица «Участие в проектах» (Job):
- create table job (  pro varchar(15)  not null  references  project (abbr),
- tabNo number(6) not null  references  emp,
- rel varchar(20)  default  'исполнитель',
- primary key (tabno, pro),
- check ( rel  IN  ('исполнитель', 'руководитель', 'консультант') )
- );

# Подмножество команд DML

- INSERT – добавление строк в таблицу.
  - Добавляет одну или несколько строк в указанную таблицу.
-
- UPDATE – изменение данных.
  - Изменяет значения одного или нескольких полей в записях указанной таблицы.
  - Можно указать условие, по которому выбираются обновляемые строки.
  - Если условие не указано, обновляются все строки таблицы.
  - Если ни одна строка не удовлетворяет условию, ни одна строка не будет обновлена.

- DELETE – удаление строк из таблицы.
  - Удаляет одну или несколько строк из таблицы.
  - Можно указать условие, по которому выбираются удаляемые строки.
  - Если условие не указано, удаляются все строки таблицы.
  - Если ни одна строка не удовлетворяет условию, ни одна строка не будет удалена.

# Добавление данных

- INSERT – добавление строк в таблицу:
- INSERT INTO имя_таблицы [(список_полей_таблицы)]
- { VALUES (список_выражений) | запрос };
- Примеры:
- -- Добавить в таблицу "Отделы" новую запись (все поля):
- insert into depart
- values(7, 'Договорной отдел');

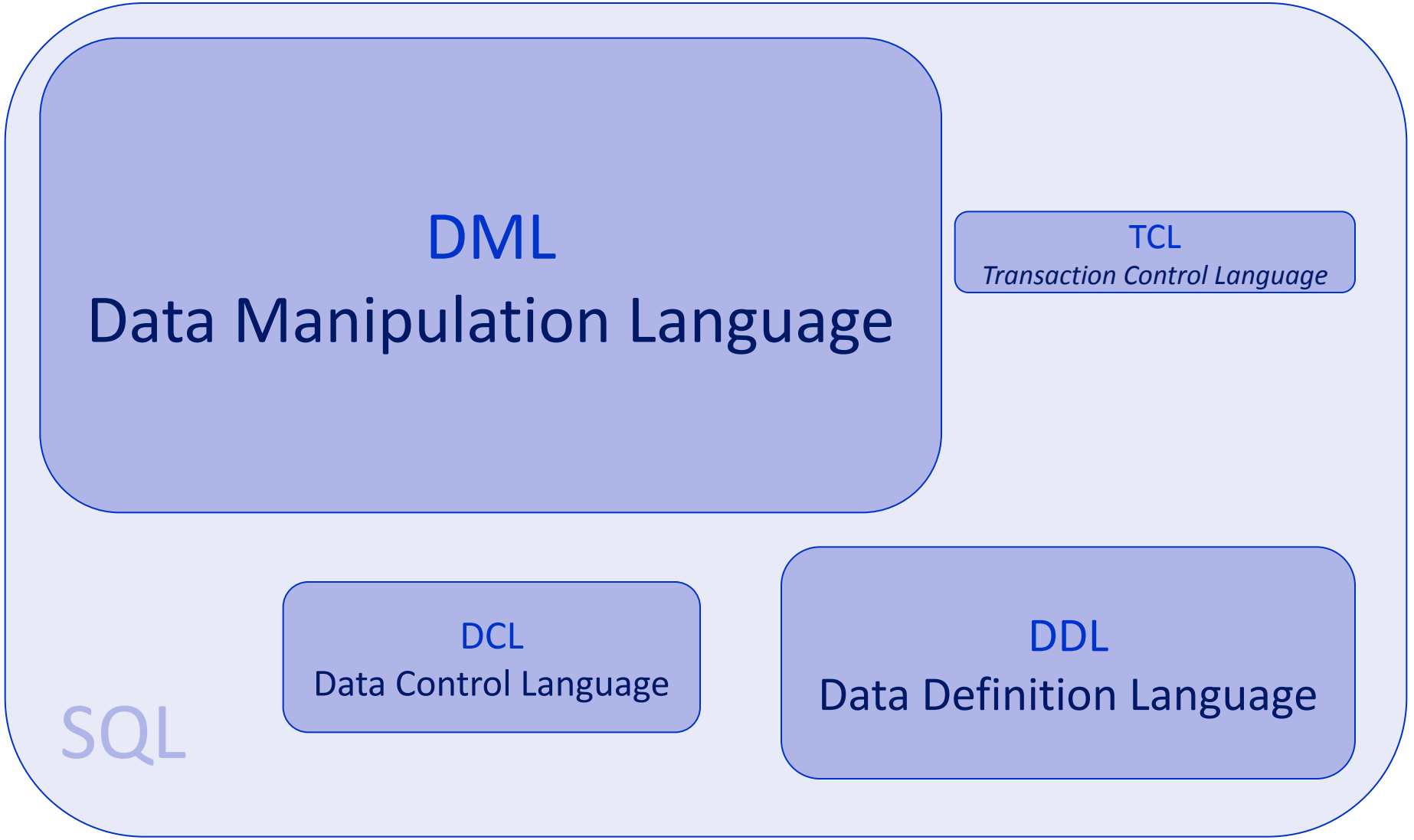
- -- Добавить в таблицу "Сотрудники" новую запись (не все поля):
- insert into emp (tabno, name, born, gender, depno, passport, pass_date_pass_get,
- post, salary, phone)
- values( 301, 'САВИН АНДРЕЙ ПАВЛОВИЧ', to_date('11.07.1969', 'dd.mm.yyyy'),
- 'М', 5, '4405092876', to_date('15.02.1999', 'dd.mm.yyyy'),
- 'ОВД "Митино" г.Москвы', 'программист', 38050, '121-34-11');
- Замечание: значение по умолчанию используется только тогда, когда значение поля не вводится в явном виде.

# Изменение данных

- UPDATE – изменение данных:
- UPDATE имя_таблицы
- SET имя_поля1 = выражение1 [, имя_поля2 = выражение2,…]
- [WHERE условие];
- Примеры:
- -- Изменить статус сотрудника Бобкова Л.П., табельный номер 74, по отношению к проекту 30."Система автоматизированного управления предприятием":
- update job
- set rel = 'консультант'
- where tabno = 74 and pro = 30;
- -- Перевести сотрудника Жаринова А.В., табельный номер 68, на должность ведущего программиста и повысить оклад на три тысячи рублей:
- update emp
- set post = 'ведущий программист', salary = salary+3000
- where tabno = 68;

# Удаление данных

- DELETE – удаление строк из таблицы:
- DELETE FROM имя_таблицы
- [ WHERE условие ];
- Примеры.
- -- Удалить сведения о том, что сотрудник Афонасьев В.Н., табельный номер 147, участвует в проектах:
- delete from job
- where tabno=147;
- -- Удалить сведения о сотруднике Афонасьеве В.Н., табельный номер 147:
- delete from emp
- where tabno = 147;
- Замечание: отменить удаление данных можно командой
- ROLLBACK;


DML
Data Manipulation Language
TCL
Transaction Control Language
DCL
Data Control Language
DDL
Data Definition Language
SQL

TCL-операторы используются для обработки транзакций.

- BEGIN [ DISTRIBUTED ] { TRAN | TRANSACTION }
  [ { trn_name | @trn_name_var } [ WITH MARK [ 'description' ] ] ] [;]

  начать транзакцию (START);

- COMMIT { [ WORK ] | [ { TRAN | TRANSACTION }
  [ trn_name | @trn_name_var ] ] } [;]

  подтвердить транзакцию;

- SAVE { TRAN | TRANSACTION }
  { save_name | @save_name_var } [;]

  установить точку отката (SAVEPOINT);

- ROLLBACK { [ WORK ] | [ { TRAN | TRANSACTION }
  [ trn_name | @trn_name_var | save_name | @save_name_var ] ] [;]

  откатить все изменения, сделанные в контексте транзакции;

- @@TRANCOUNT, XACT_ABORT, XACT_STATE()

  системные переменные, параметры и функции.

Пример управления транзакцией:

```
USE AdventureWorks2008R2;

CREATE TABLE Test(id INT);

BEGIN TRANSACTION;

  INSERT INTO Test(id) VALUES(1);
  INSERT INTO Test(id) VALUES(2);
  UPDATE Test SET id=200 WHERE id=1;

SAVE TRANSACTION s_name;

  UPDATE Test SET id=1000 WHERE id=2;

ROLLBACK TRANSACTION s_name;

  SELECT id FROM Test;

DROP TABLE Test;
```

Проблемы параллельного доступа к данным:

- потерянное обновление;
- «грязное» чтение;
- неповторяющееся чтение;
- фантомное чтение – отличается от предыдущего тем, что данные не изменяются/удаляются, а добавляются новые (фантомные) записи.

Уровни изоляции транзакций:

- неподтверждённое чтение (read uncommitted, dirty read) — чтение незафиксированных всех транзакций – гарантирует только отсутствие потерянных обновлений;
- подтверждённое чтение (read committed) — чтение зафиксированных изменений параллельных транзакций;
- повторяемое чтение (repeatable read, snapshot) — все изменения параллельных транзакций после начала своей недоступны;
- упорядоченный (serializable) — все транзакции выполняются строго последовательно.

Типы транзакций:

- явная (explicit) – транзакция начинается оператором начала транзакции или вызовом API-функции;
- автоматическая (autocommitted) – режим по-умолчанию – каждый оператор автоматически начинает транзакцию и подтверждает ее;
- пакетная (batch-scoped) – в режиме MARS.

Примечание: технологию MARS следует использовать с осторожностью, т.к. при переключении транзакций в явный режим с помощью API, операторы COMMIT и ROLLBACK приведут к откату всего пакета.

Требования к транзакциям (ACID):

- атомарность (atomicity);
- согласованность (consistency);
- изолированность (isolation);
- долговечность (durability).

DCL-операторы используются управления доступом к объектам СУБД, базы данных и к отдельным операторам SQL.

- GRANT { [ ALL [ PRIVILEGES ] ] | permission [ ( column [ ,...n ] ) ] [ ,...n ] }
  [ ON [ class :: ] securable ] TO principal [ ,...n ]
  [ WITH GRANT OPTION ] [ AS principal ]

  предоставление разрешения на определенное действие с объектом;

- DENY { [ ALL [ PRIVILEGES ] ] | permission [ ( column [ ,...n ] ) ] [ ,...n ] }
  [ ON [ class :: ] securable ] TO principal [ ,...n ]
  [ CASCADE] [ AS principal ]

  устанавливает запрет на действие с объектом;

- REVOKE [ GRANT OPTION FOR ]
  { [ ALL [ PRIVILEGES ] ] | permission [ ( column [ ,...n ] ) ] [ ,...n ] }
  [ ON [ class :: ] securable ] { TO | FROM } principal [ ,...n ]
  [ CASCADE] [ AS principal ]

  удаляет разрешение или запрет;

Примечание: DENY превалирует над GRANT (в большинстве случаев).

## Примеры управления разрешениями:

```
USE AdventureWorks2008R2;

GRANT SELECT ON OBJECT::Person.Address TO RosaQdM;

GRANT REFERENCES (BusinessEntityID) ON OBJECT::HumanResources.vEmployee TO
Wanida WITH GRANT OPTION;

DENY EXECUTE ON OBJECT::HumanResources.uspUpdateEmployeeHireInfo
   TO Recruiting11;

DENY EXECUTE ON XML SCHEMA COLLECTION::Sales.Invoices4
TO Wanida;

REVOKE IMPERSONATE ON LOGIN::WanidaBenshoof FROM [AdvWorks\YoonM];

REVOKE VIEW DEFINITION ON ENDPOINT::Mirror7 FROM ZArifin;
```

DDL-операторы используются для создания, изменения и удаления объектов СУБД или базы данных.

- CREATE – создает объект;
- ALTER – изменяет существующий объект или составные части его;
- DROP – удаляет объект;
- TRUNCATE – очищает таблицу.

Стандарт SQL-92 определяет команду CREATE в вариантах: ASSERTION, CHARACTER SET, COLLATION, DOMAIN, SCHEMA, TABLE, TRANSLATION, VIEW.

В MySQL 5.1 – 12 вариантов, в SQL Server 2008 R2 – 59 вариантов.

Примечание: с помощью системных объектов следует проверять существование объектов СУБД или базы данных.

## Пример оператора создания таблицы:

```
USE AdventureWorks2008R2;

CREATE TABLE dbo.PurchaseOrderDetail
(
   PurchaseOrderID int NOT NULL
      REFERENCES Purchasing.PurchaseOrderHeader(PurchaseOrderID),
   LineNumber smallint NOT NULL,
   ProductID int NULL
      REFERENCES Production.Product(ProductID),
   UnitPrice money NULL,
   OrderQty smallint NULL,
   ReceivedQty float NULL,
   RejectedQty float NULL,
   DueDate datetime NULL,
   rowguid uniqueidentifier ROWGUIDCOL  NOT NULL
      CONSTRAINT DF_PurchaseOrderDetail_rowguid DEFAULT (newid()),
   ModifiedDate datetime NOT NULL
      CONSTRAINT DF_PurchaseOrderDetail_ModifiedDate DEFAULT (getdate()),
   LineTotal  AS ((UnitPrice*OrderQty)),
   StockedQty  AS ((ReceivedQty-RejectedQty)),
   CONSTRAINT PK_PurchaseOrderDetail_PurchaseOrderID_LineNumber
         PRIMARY KEY CLUSTERED (PurchaseOrderID, LineNumber)
         WITH (IGNORE_DUP_KEY = OFF)
)
ON PRIMARY WITH (DATA_COMPRESSION = PAGE);
```

DML-операторы используются для манипулирования данными: выборки, вставки, удаления или изменения данных.

- SELECT – возвращает набор данных;
- UPDATE – изменяет существующие данные;
- INSERT – добавляет новые данные;
- MERGE – слияние наборов данных;
- DELETE – удаляет данные.

CRUD-операции: create, read (retrieve), update (modify) and delete (destroy). Часто используется совместно с термином «DML-операторы», а иногда и подменяет его.

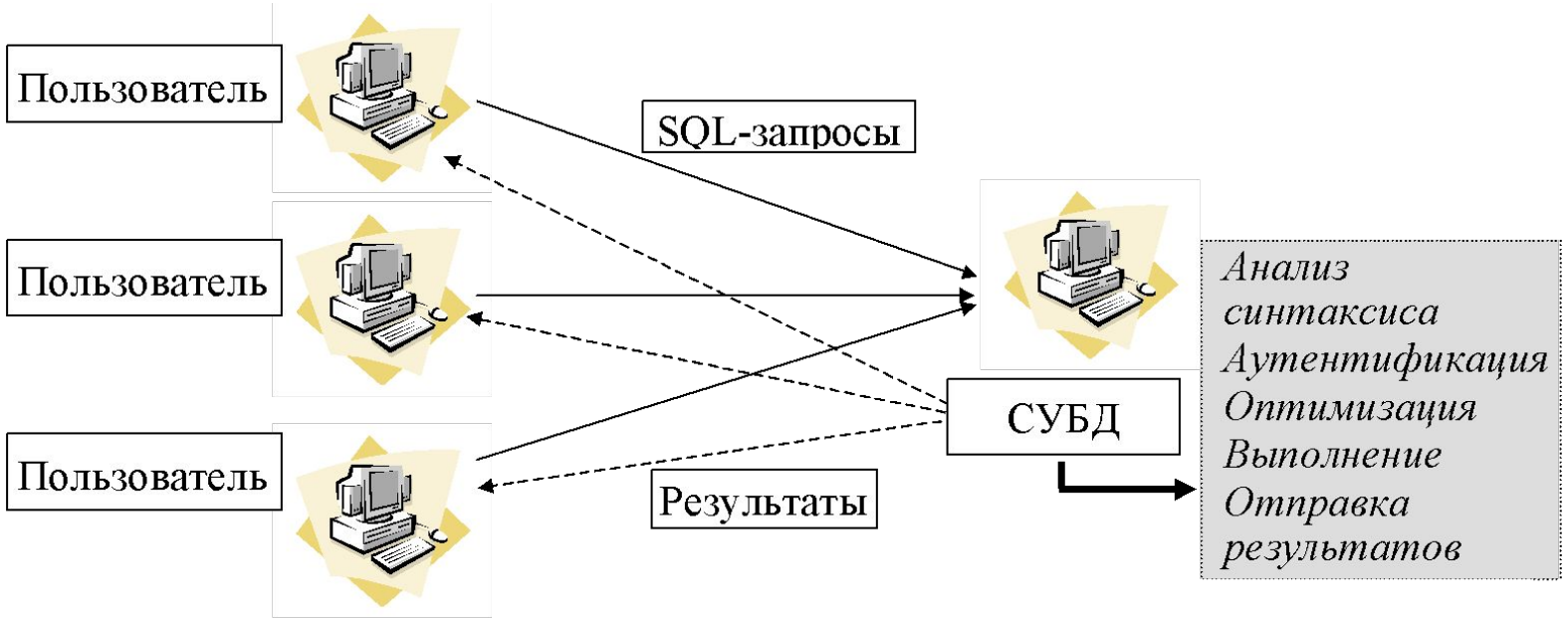
CRUD – термин компьютерной науки, и определяется как минимальный достаточный набор функций постоянного хранилища данных.

## Пользователь

- устанавливает соединение с БД;
- вводит команду SQL;
- инициирует выполнение команды.

## СУБД

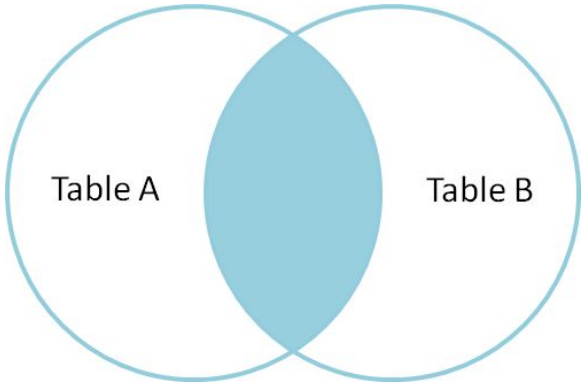
- выполняет синтаксический анализ запроса;
- проверяет наличие прав на выполнение этого запрос;
- выбирает план выполнения запроса;
- выполняет запрос;
- результат выполнения отсылает пользователю.

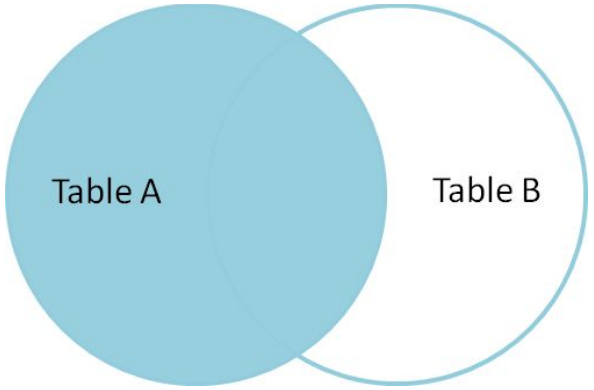


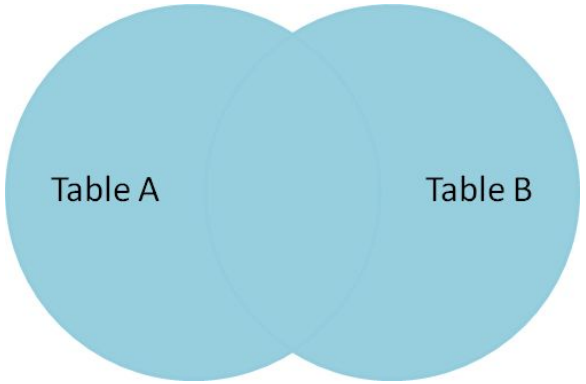
Полный синтаксис оператора SELECT очень сложный, однако в нем можно выделить следующие блоки:

- WITH <common_table_expression> – блок задания общего табличного выражения;
- SELECT select_list – блок задания столбцов результирующего набора;
- FROM table_source – блок задания источников данных;
- WHERE search_condition – блок условий отбора;
- GROUP BY group_by_expression – блок задания столбцов группировки / агрегирования данных;
- HAVING search_condition – блок условий отбора агрегированных;
- ORDER BY order_expression [ ASC | DESC ] – блок сортировки набора данных;
- { UNION | EXCEPT | INTERSECT } select_query – блок множественных операций нескольких наборов данных.

# Демонстрация соединения таблиц



# Демонстрация примеров SELECT

```
UPDATE Sales.SalesPerson
SET SalesYTD = SalesYTD + SubTotal
FROM Sales.SalesPerson AS sp
      JOIN Sales.SalesOrderHeader AS so
            ON sp.BusinessEntityID = so.SalesPersonID
                  AND so.OrderDate = (SELECT MAX(OrderDate)
                                      FROM Sales.SalesOrderHeader
                                      WHERE SalesPersonID = sp.BusinessEntityID);

UPDATE Sales.SalesPerson
SET SalesYTD = SalesYTD +
   (SELECT SUM(so.SubTotal)
    FROM Sales.SalesOrderHeader AS so
    WHERE so.OrderDate = (SELECT MAX(OrderDate)
                          FROM Sales.SalesOrderHeader AS so2
                          WHERE so2.SalesPersonID = so.SalesPersonID)
    AND Sales.SalesPerson.BusinessEntityID = so.SalesPersonID
    GROUP BY so.SalesPersonID);
```

# Демонстрация примеров UPDATE, INSERT, MERGE, DELETE

Системы управления базами данных (СУБД).

Установка, настройка СУБД.

Примеры различных СУБД.

СУБД - совокупность программного обеспечения, обеспечивающего создание и использование БД, хранение, чтение и контроль данных, а так же контроль доступа пользователей к БД и данным.








- Управление данными на внешних носителях (жесткие диски, ленты и др.).
- Управление данными в оперативной памяти.
- Журналирование изменений, резервное копирование и восстановление БД после аварий.
- Поддержка лингвистических средств определения и манипулирования данными.

## По способу доступа к данным

- Файл-серверные. Сервер лишь место хранения, обработка на клиенте, файловые блокировки, нагрузка на сеть, мощные пользовательские ПК, сбой клиента ведет к краху.
  MS Access, Paradox, dBase, FoxPro

- Клиент-серверные. Обработка на сервере, централизаций управления, слабые пользовательские ПК.
  MS SQL Server, Oracle Database, MySQL.

- Встраиваемые. Часть готового продукта, мобильные устройства, единое пространство с приложением.
  MS SQL Server Compact, OpenEdge, SQLite, Firebird Embedded, InterBase SMP

## По масштабам задач

- Настольные. Файл-серверный доступ, ограниченный функционал, до 20 пользователей, малый бизнес.
  dBase, FoxPro, MS Access.

- Серверные. Клиент-сервер(n-звеньев), высокие нагрузки, крупный бизнес, обработка на сервере, централизаций управления, слабые пользовательские ПК.
  MS SQL Server, Oracle Database, DB2.

## По модели данных (как БД)

- Реляционные: MS SQL Server, Oracle Database, MySQL.
- Объектные: Cache, GemStone.
- Иерархические: IMS от IBM, System 2000 от SAS-Institute.
- Сетевые: Cerebrum, dbVista.

- Выбор аппаратной платформы.

- Установка необходимого системного ПО.

- Выбор требующихся для установки компонент СУБД.

- Выбор размещения каталогов файлов БД, журналов, системного каталога.

- Конфигурация запуска служб СУБД.

- Конфигурация сетевой доступности.

Настройка требуется не всегда. Прежде чем настраивать, нужно определить цель.

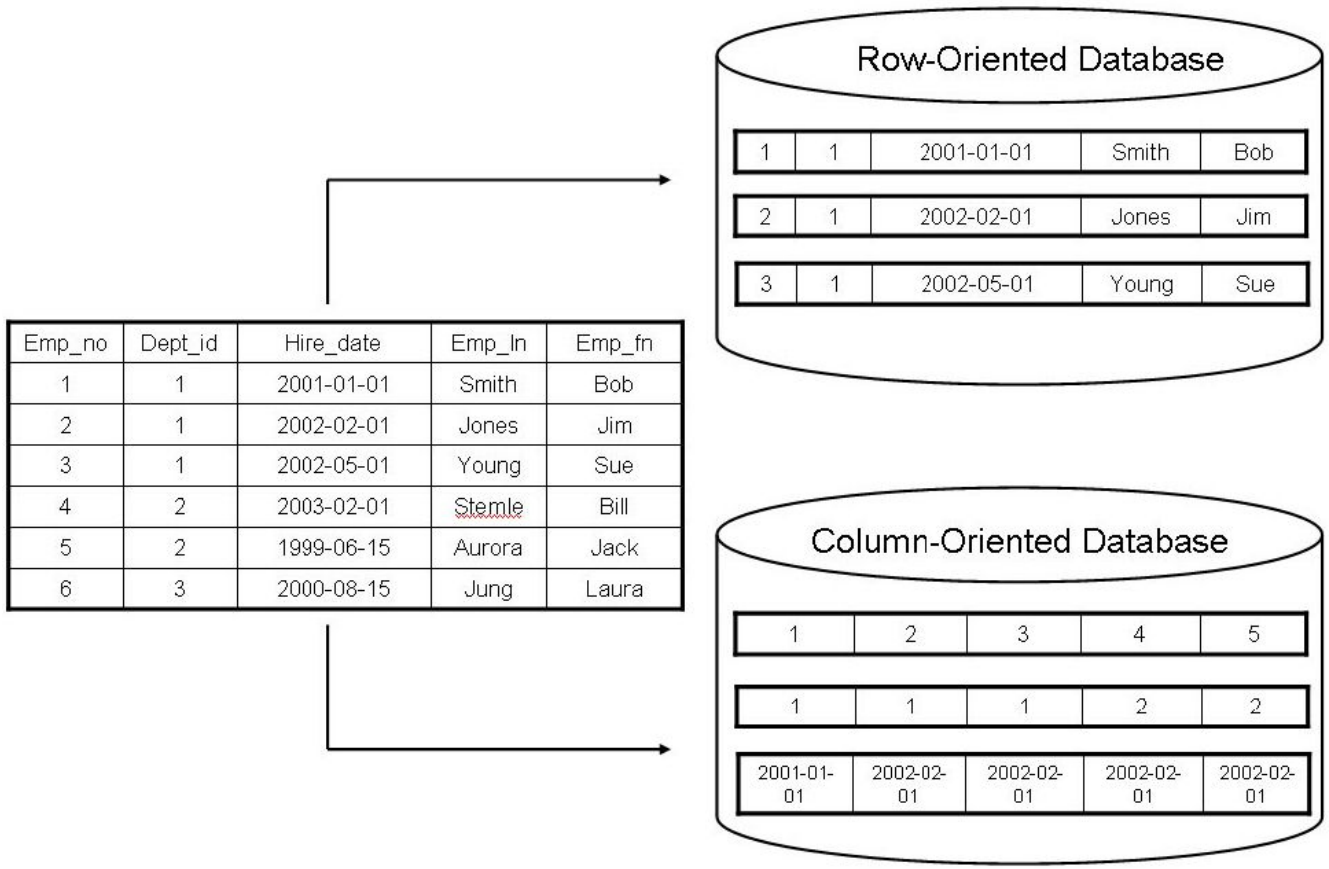
## Виды настроек:

- Настройка инфраструктуры (резервное копирование, импорт данных, сбор статистики, переиндексация, событийная активность, права доступа).
- Настройка производительности (время отклика, время выполнения, выравнивание загрузки сервера).

## Средства настройки:

- Переписывание кода «тяжелых» запросов.
- Настройка инфраструктуры (размещение данных, индексы, кэширование, директивы оптимизатору).
- Наращивание аппаратной мощи (коэффициент < 1).

Хранят данные не построчно, а по столбцам.
Эффективны в аналитических системах с преобладающими операциями чтения




| Emp_no | Dept_id | Hire_date | Emp_ln | Emp_fn |
|---|---|---|---|---|
| 1 | 1 | 2001-01-01 | Smith | Bob |
| 2 | 1 | 2002-02-01 | Jones | Jim |
| 3 | 1 | 2002-05-01 | Young | Sue |
| 4 | 2 | 2003-02-01 | Stemle | Bill |
| 5 | 2 | 1999-06-15 | Aurora | Jack |
| 6 | 3 | 2000-08-15 | Jung | Laura |

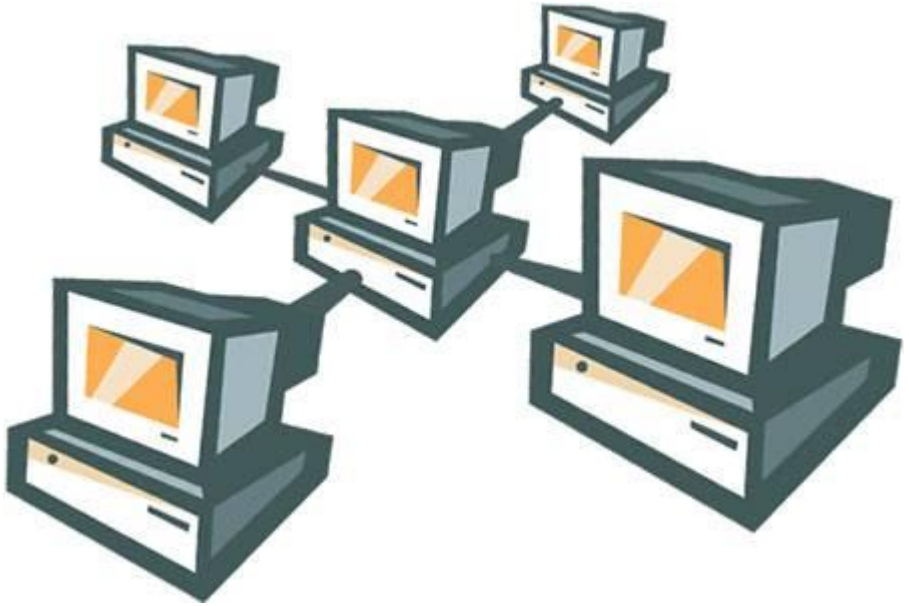
# Обеспечение безопасности и контроль доступа к базе данных.

Абсолютно безопасное ПО – недостижимая цель.

ПО должно быть достаточно безопасным.

Нет одного регламента обеспечения безопасности, но есть много действий, которые сводят безопасность к нулю.

Безопасность системы равна безопасности самого слабого ее места.



- Пользователь должен иметь только одну учетную запись. Нет обезличенным учеткам, нет передаче паролей, подменяешь – получи временные права.

- Смена пароля каждые 3 - 6 месяцев.

- Длина пароля должна быть не менее 6 символов. Вариантов $4*10^{12}$

- Количество попыток входа в систему не должно превышать 5. Блокировка узла, блокировка учетки, временное ограничение активности.

## Встроенные средства.

Создается отдельная учетка на MS SQL, он всё и контролирует. Web-сервисы.

```
CREATE LOGIN BillGates
WITH
PASSWORD = 'iHateRipeApples'
MUST_CHANGE,
DEFAULT_DATABASE =
AdventureWorks,
DEFAULT_LANGUAGE = Russian,
CHECK_EXPIRATION = ON,
CHECK_POLICY =  ON
```

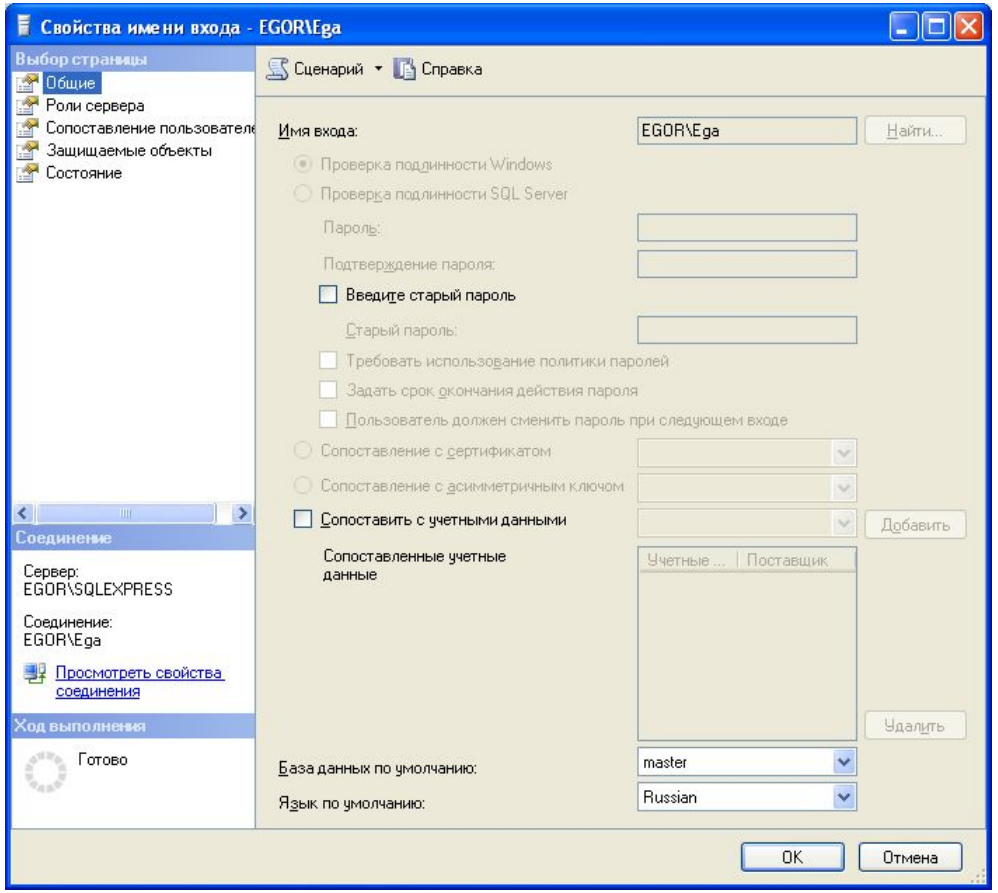
## Интеграция с учетными данными Windows.

Учетные данные берутся из домена Windows: из пользователей и из групп. Корпоративные приложения.

```
CREATE LOGIN [Microsoft\BillGates]
FROM WINDOWS
WITH DEFAULT_DATABASE =
AdventureWorks,
DEFAULT_LANGUAGE = Russian
```

# ALTER LOGIN

```
ALTER LOGIN <login name>
[ { ENABLE | DISABLE } ]
[ { WITH
      PASSWORD = '<password>'
      [{[UNLOCK ] [MUST_CHANGE]}
      DEFAULT_DATABAS E =
      <database>
      DEFAULT_LANGUAGE =
      <language>
      NAME = <new login name>
      CHECK_EXPIRATION
      ={ON|OFF}
      CHECK_POLICY
            ={ON|OFF}}]
```



Управление доступом пользователей к БД начинается когда уже имеется имя входа на сервер. Включает в себя создание пользователя базы данных (USER) и предоставление ему прав на выполнение определенных действии над определенными объектами БД.

Создание пользователя базы данных выполняется из программы Management Studio или с помощью команды «CREATE USER»

```
CREATE USER [Microsoft\BillGates]
    FOR LOGIN [Microsoft\BillGates]
    WITH DEFAULT_SCHEMA = [sales]
```

Без тонкой настройки прав доступа на выполнение конкретных операций над конкретными объектами БД невозможно обеспечить гибкую систему безопасности. Deny сильнее GRANT.

| | | | | | |
|---|---|---|---|---|---|
| *GRANT ** | *SELECT* | *ON* | *[Product]* | *TO* | *[MyBDUser]* |
| *DENY *** | *INSERT* | | | | |
| *REVOKE* | *UPDATE* | | | | |
| | *DELETE* | | | | |
| | *EXECUTE* | | | | |

** [WITH GRANT OPTION]*
*** [CASCADE]*

Разрешения могут быть даны не только на объекты базы данных, но и на операторы, которые не связаны с конкретными объектами БД. Такие операторы обеспечивают управление безопасностью на уровне сервера.

*CREATE DATABASE* – право создавать базы данных.

Колоночные СУБД

*CREATE TABLE* - право создавать таблицы.

*CREATE VIEW* – право создавать представления.

*CREATE PROCEDURE* – право создавать хранимые процедуры.

*BACKUP DATABASE* – право создавать резервные копии БД.

Роль — это совокупность прав доступа, которые можно сразу же назначить пользователю, определив его как члена этой роли.

Пользователь может принадлежать к нескольким ролям. Роль позволяет группировать права доступа в логические группы, а затем применять различные их сочетания, создавая наборы прав, наиболее подходящие для конкретного пользователя.

Роли подразделяются на две категории:
- роли сервера (фиксированные),
- роли базы данных (фиксированные+ пользовательские).

Роль sysadmin - любые действия. Win Administrators по умолчанию члены.
Роль serveradmin - конфигурировать сервер, останавливать работу сервера.
Роль setupadmin ограничивается управлением связанными серверами и процедурами запуска.
Роль processadmin предоставляет возможность управлять процессами, уничтожать процессы.
Роль dbcreator – создание и модификацией баз данных
Роль diskadmin позволяет управлять файлами, присоединять и отсоединять базы данных.
Роль bulkadmin - выполнение оператора массовой вставки «BULK INSERT», но не дает прав на оычный «INSERT»

Роль db_owner - владелец БД, обычно полные права.
Роль db_accessadmin - добавление пользователей БД.
Роль db_datareader - доступ к выборке из всех таблиц БД.
Роль db_datawriter - изменение данных всех таблиц БД.
Роль db_ddladmin - разрешено выполнение DDL операторов.
Роль db_securityadmin – не создает пользователей в БД, но позволяет управлять ролями и членами ролей, определять права на выполнение операторов и права доступа к объектам.
Роль db_backupoperator - выполнять резервное копирование.
Роль db_denydatareader - эквивалентно оператору «DENY SELECT», применительно к каждой таблице и представлению в базе данных.
Роль db_denydatawriter аналогична роли db_denydatareader, но распространяется только на операторы «INSERT», «UPDATE», «DELETE»

Являются реальной основой системы обеспечения безопасности базы данных. При создании этих ролей необходимо принять решение о том, какие права они должны включать.

Дают возможность классифицировать пользователей по категориям доступа, поскольку роли позволяют вносить изменения в одном месте, а затем экстраполировать эти изменения по всем членам роли.

```
CREATE ROLE [TestRole]
GRANT SELECT ON [Product] TO [TestRole]
EXECUTE sp_addrolemember
                    @rolename = [TestRole],
                    @membername = [MyAccount]
```

- Настройте порты TCP\IP

О стандартном порте 1433 знают все – замените.

- Держите sa под замком

Создайте невообразимый пароль, не давайте его пользователям.

- Используйте представления, хранимые процедуры и функции для обеспечения безопасности.

Это скрывает структуру БД. Представления и табличные функции скрывают ширину и глубину данных. Хранимые процедуры скрывают логику и позволяют манипулировать данными, не предоставляя на них доступ.

# Microsoft Access

Технология создания реляционной базы данных (РБД)

# Этапы проектирования РБД

- Построение информационно-логической модели данных предметной области
- Определение структуры РБД
- Конструирование таблиц БД в Access
- Создание схемы данных в Access
- Ввод данных в таблицы (создание записей)

# Информационно-логическая модель данных

**Информационно-логическая модель (ИЛМ) отображает данные предметной области в виде совокупности информационных объектов и связей между ними.**
**Примерами информационных объектов могут быть: ТОВАР, ПОСТАВЩИК, ЗАКАЗЧИК, СОТРУДНИК, ПОСТАВКА**

# Виды информационных объектов РБД

- Справочные (список сотрудников, прайс-лист, список категорий изделий, нормативы)
- Учетно-отчетные (отражают сведения о заказах, выполненных работах, произведенной продукции)

# Связи информационных объектов

Связь  устанавливается между двумя логически взаимосвязанными информационными объектами, например:

- Поставщик - товар
- Склад - готовая продукция
- Группа - студент

# Виды информационных связей между объектами РБД

- Одно-однозначные 1:1(каждому экземпляру первого объекта соответствует один экземпляр второго)
- Одно-многозначные 1:M (каждому экземпляру первого объекта соответствует несколько экземпляров второго)
- Много-многозначные M:N (каждому экземпляру первого объекта соответствует несколько экземпляров второго и наоборот, (каждому экземпляру второго объекта соответствует несколько экземпляров первого) )

# Подчиненность связанных объектов

В паре связанных объектов 1:1 и 1:М один объект является главным, а другой – подчиненным.
Главный объект обычно содержит справочную информацию, а подчиненный - учетно- отчетную.

# Логическая структура РБД

Каждый объект информационно-логической модели отображается реляционной таблицей. Каждый столбец (поле) реляционной таблицы соответствует одному из реквизитов объекта. Одно из полей определяется как КЛЮЧЕВОЕ. В каждой паре реляционных таблиц должно быть хотя бы одно одинаковое поле для связи.

# Пример связей 1:1 и 1:М между таблицами РБД

# Пример связи M:M между таблицами РБД


Связь M:M между таблицами Авторы и Книги реализуется в РБД с помощью таблицы-связки Книги/Авторы

# Порядок создания РБД

- Создать таблицы в режиме Конструктор
- Установить связи между таблицами
- Создать формы для таблиц
- Заполнить таблицы РБД через формы

# Роль связей между таблицами РБД

- Позволяют иерархически просматривать связанные записи из всех таблиц
- Дают возможность автоматической выборки данных, относящихся к одному объекту, из всех таблиц
- Позволяют контролировать правильность действий пользователя при добавлении и удалении записей

# Просмотр связанных записей

# Организация связи между таблицами

Устанавливать связь между одноименными полями двух реляционных таблиц, проводя линию связи от КЛЮЧЕВОГО поля ГЛАВНОЙ таблицы к одноименному полю ПОДЧИНЕННОЙ Какая из таблиц  главная должен определять пользователь. В процессе создания связей 1:1 и 1:М необходимо задавать ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ

# Обеспечение целостности данных в РБД

Обеспечение целостности данных означает выполнение для взаимосвязанных таблиц следующих условий корректировки БД:

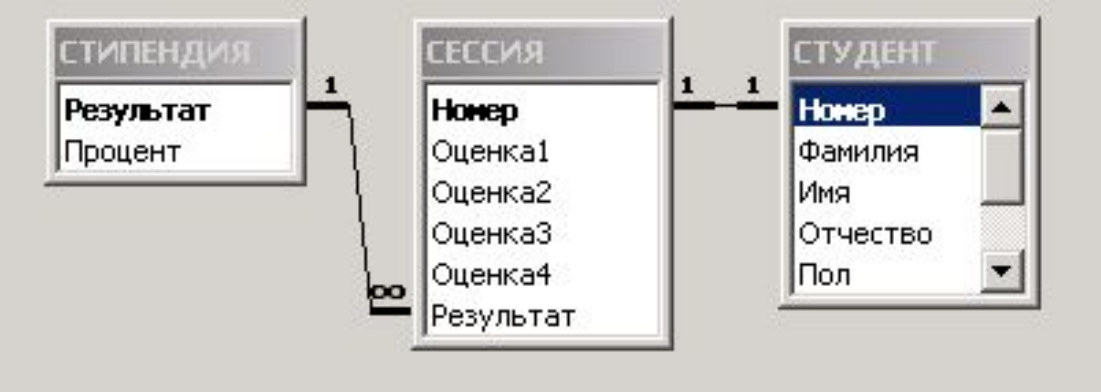
- 1. В подчиненную таблицу не может быть добавлена запись с не существующим в главной таблице значением ключевого поля;

  2. В главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;

- 3. Изменение значений ключа связи главной таблицы должны приводить к изменению соответствующих значений в записях подчиненной таблицы.

Если установлен только параметр "Обеспечение целостности данных", то при попытке нарушить это условие Access выдает предупреждение. Если установлены параметры каскадного обновления и удаления записей, то Access будет автоматически производить корректировку данных в связанных таблицах.

# Обеспечение целостности данных. Пример




В таблицу Сессия нельзя ввести запись со значением поля НОМЕР, которого нет в главной таблице Студент. Удаление записи в главной таблице Студент приведет к автоматическому удалению связанной записи в таблице Сессия.

**РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ РЕШЕНИЯ ЗАДАЧ НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ С ИСПОЛЬЗОВАНИЕМ СУБД ACCESS**

# Курсовая работа

Выполнил: студент
группы 08-ПИ

Проверил доцент
Лобова О.Е.

г.Сочи, 2010

# Цель создания БД

- Разработка средствами приложения MS Access автоматизированной системы «Начисление зарплаты» для малого предприятия.

# Для реализации поставленной цели надо решить следующие задачи:

- Изучить предметную область
- Выбрать СУБД.
- Построить информационно-логическую модель.
- Реализовать информационно-логическую модель с использованием СУБД. (Создать БД по информационно-логической модели)
- Создать пользовательский интерфейс.
- Создать инструкцию пользователя

# Актуальность

- Расчет заработной платы сотрудникам малых предприятий производиться бухгалтерами либо с помощью программы «1С-бухгалтерия», либо вручную. Так как программа «1С-бухгалтерия» очень сложна в применении, и ее может освоить не каждый бухгалтер, то расчет заработной платы производится с помощью электронных таблиц Excel.. В данной работе будут рассмотрены принципы создания информационной системы «Начисление заработной платы» для малого предприятия с помощью СУБД MS Access, ориентированной на комфортную работу бухгалтеров.

# Структура БД

- Сведения о сотрудниках: Ф.И.О., подразделение, должность, оклад, количество детей и т.п.
- Сведения о премиях, надбавках и мат. помощи.
- Табель учета рабочего времени.

Пр

Унифицированная форма № Т-2
Утверждена постановлением
Госкомстата
России от 06.04.01 № 26

| | Код |
|---|---|
| Форма по ОКУД | 0301002 |
| по ОКПО | |

_____
(Наименование организации)

| Номер документа | Дата составления | Отчетный период | |
|---|---|---|---|
| | | с | по |
| | | | |

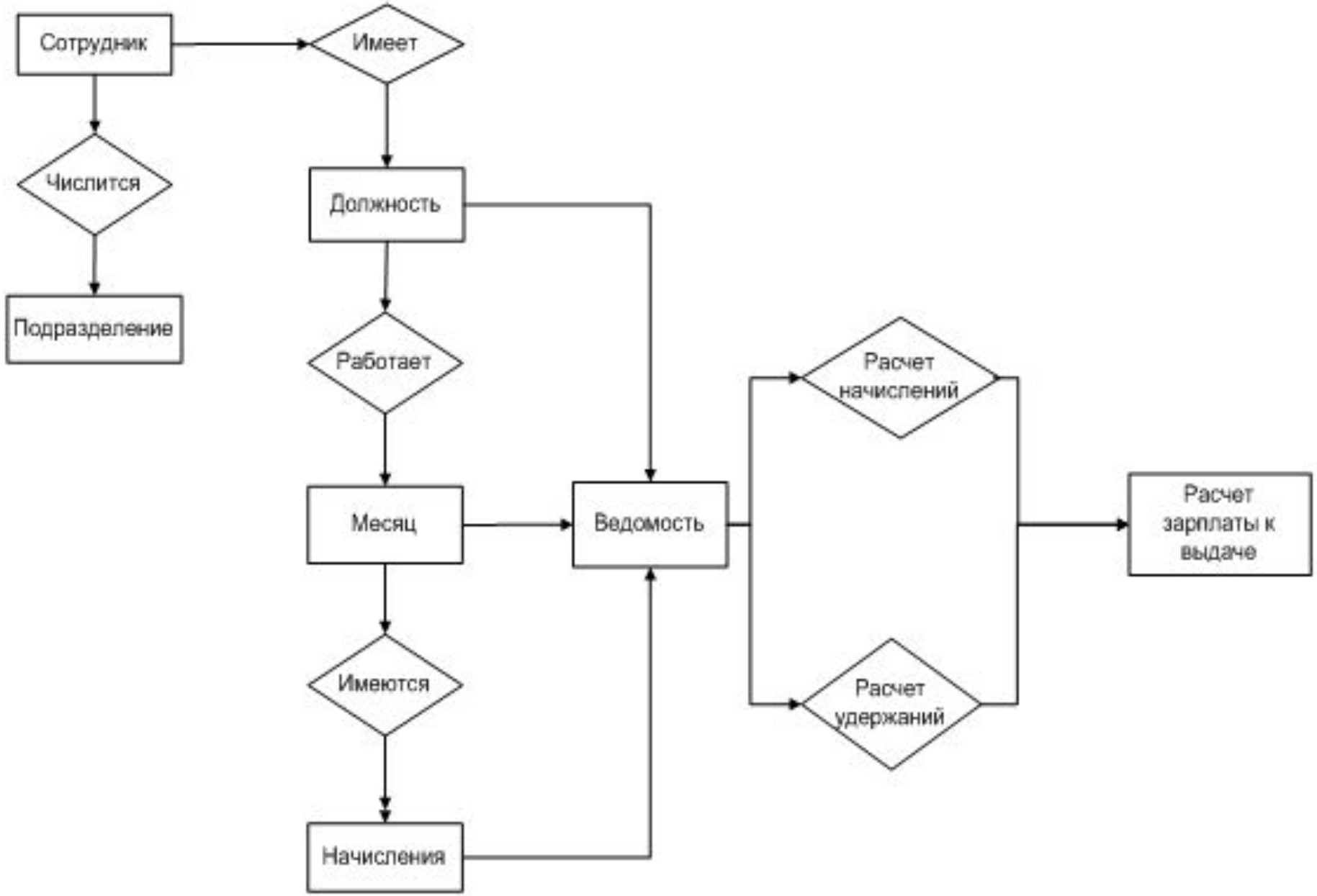
**Т А Б Е Л Ь**
**учета рабочего времени**
**и расчета оплаты труда**

| № п/п | Таб. номер | Ф.И.О. | Профессия, (должность) | Кол-во неявок | Отработано дней (рабочих, выходных, праздничных) | Кол-во рабочих дней |
|---|---|---|---|---|---|---|
| | | | | | | |

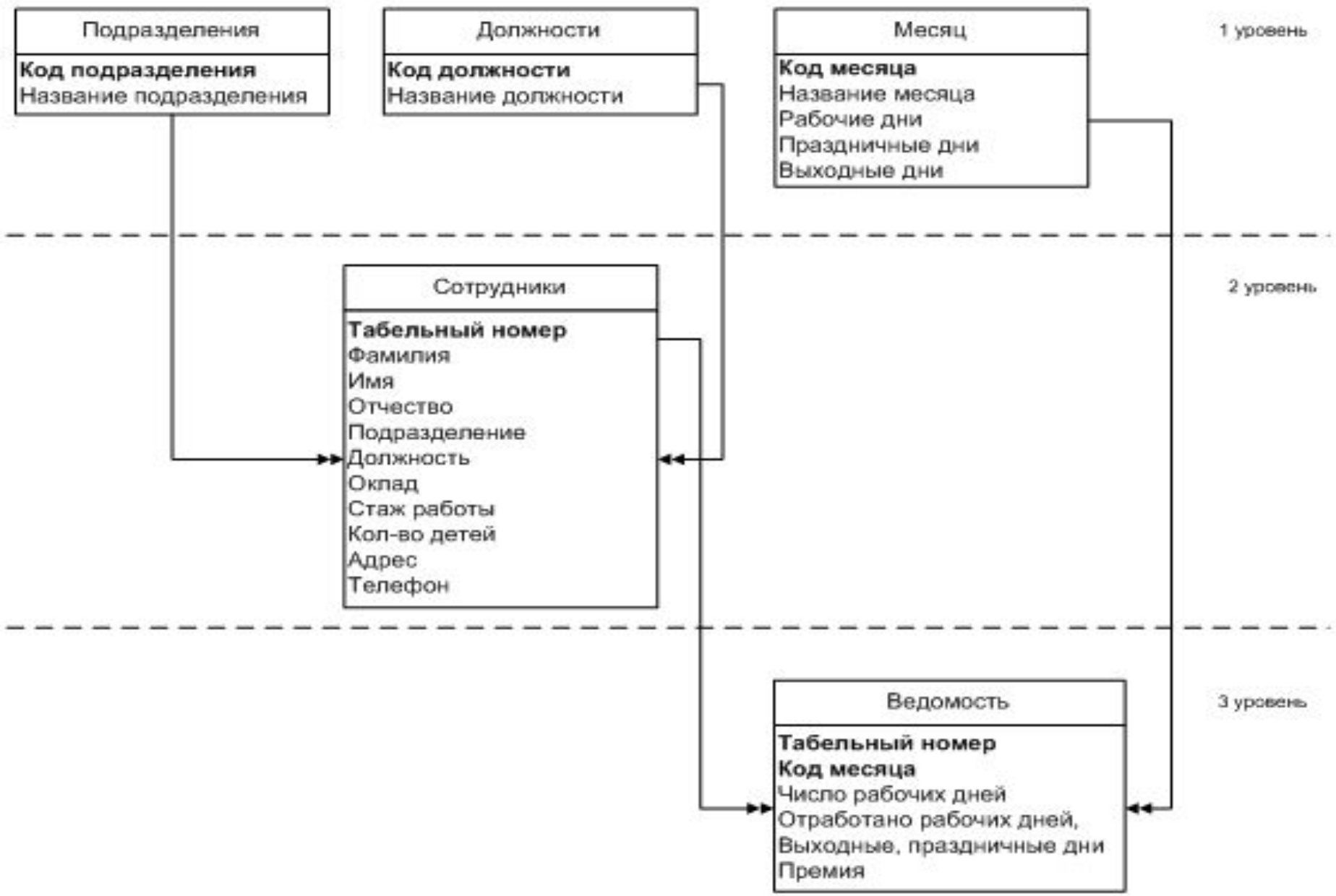
**Руководитель структурного подразделения** _____ _____ _____
должность подпись расшифровка подписи

**Работник** _____
подпись

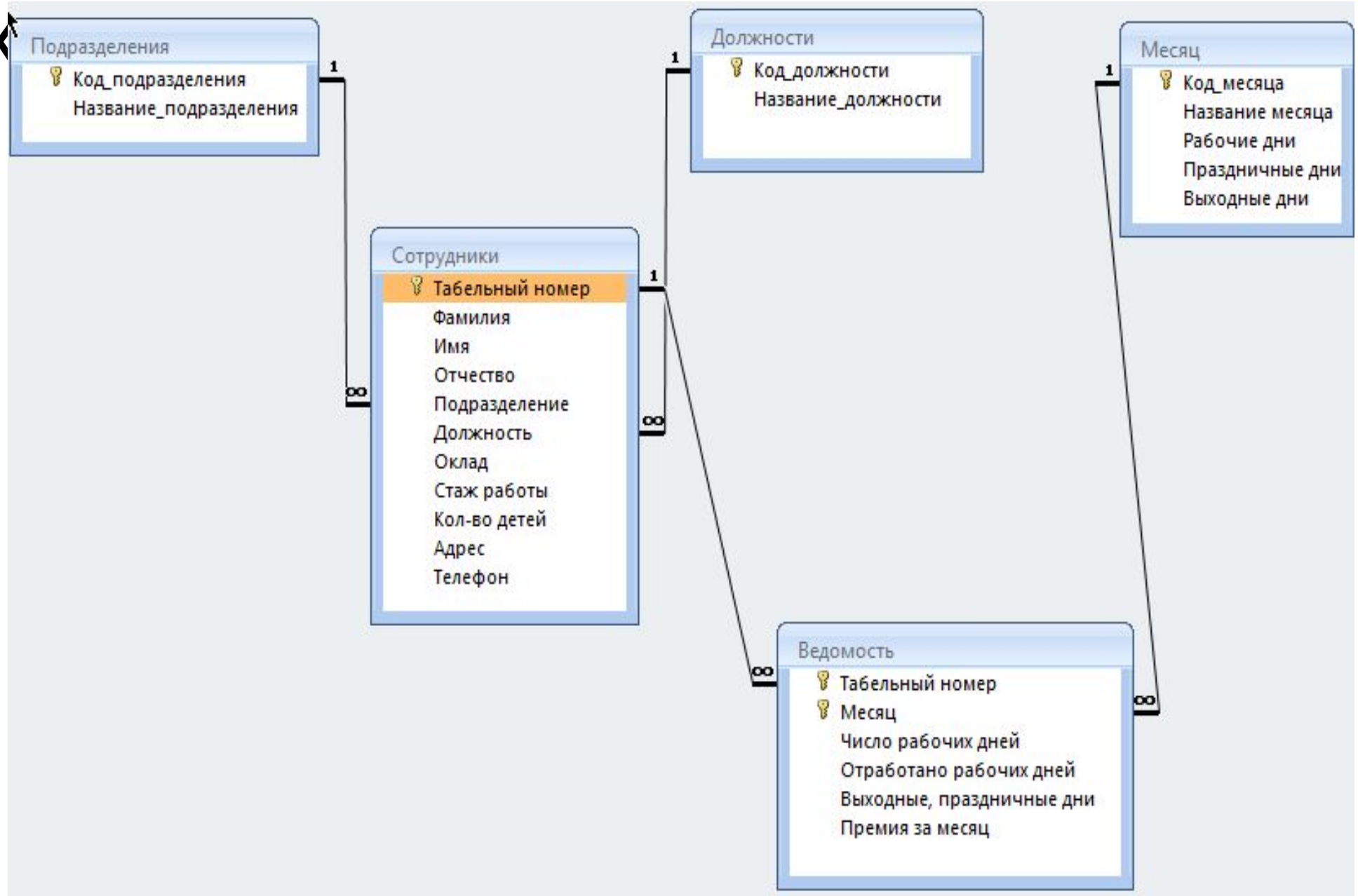
ER »


Сотрудник
Имеет
Числится
Должность
Подразделение
Работает
Расчет начислений
Месяц
Ведомость
Расчет зарплаты к выдаче
Имеются
Расчет удержаний
Начисления

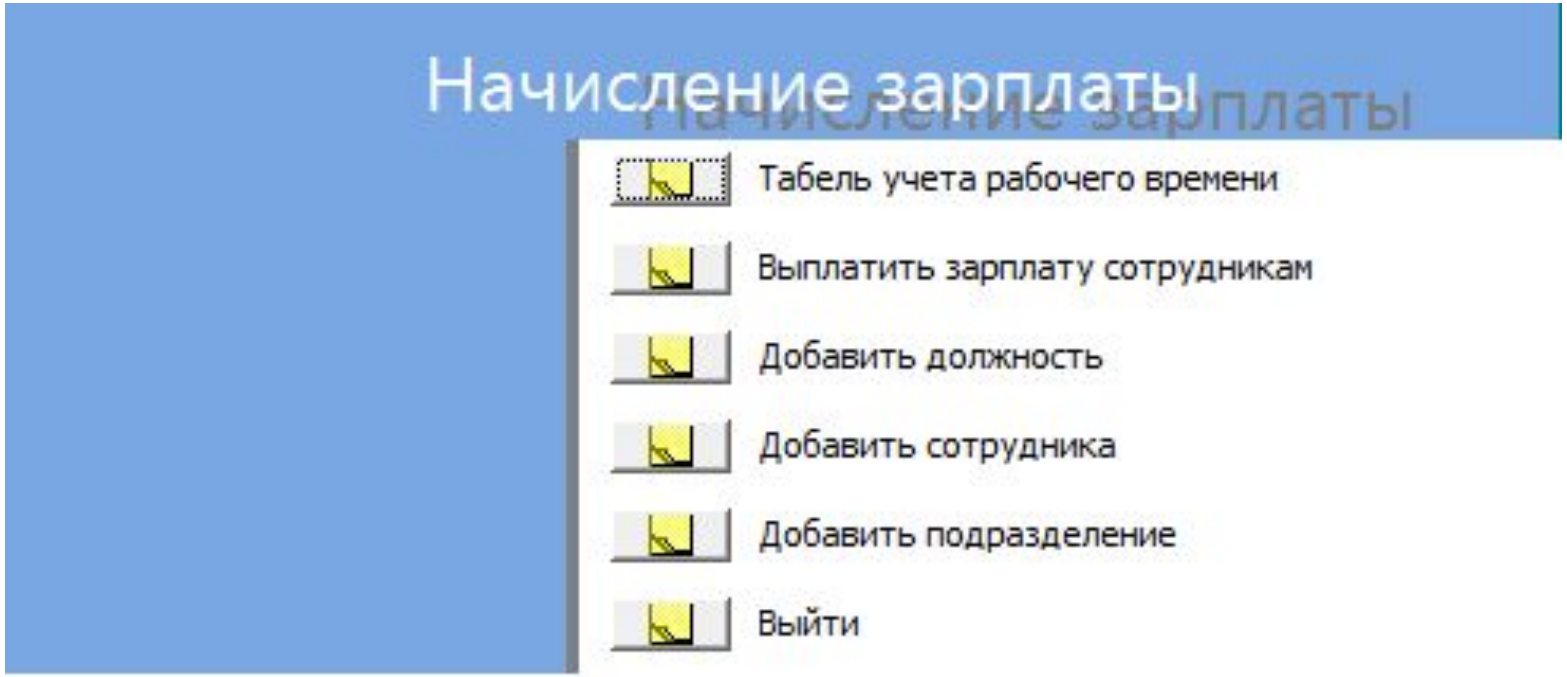
ИЛ


Подразделения
Код подразделения
Название подразделения
Должности
Код должности
Название должности
Месяц
Код месяца
Название месяца
Рабочие дни
Праздничные дни
Выходные дни
1 уровень
Сотрудники
Табельный номер
Фамилия
Имя
Отчество
Подразделение
Должность
Оклад
Стаж работы
Кол-во детей
Адрес
Телефон
2 уровень
Ведомость
Табельный номер
Код месяца
Число рабочих дней
Отработано рабочих дней,
Выходные, праздничные дни
Премия
3 уровень

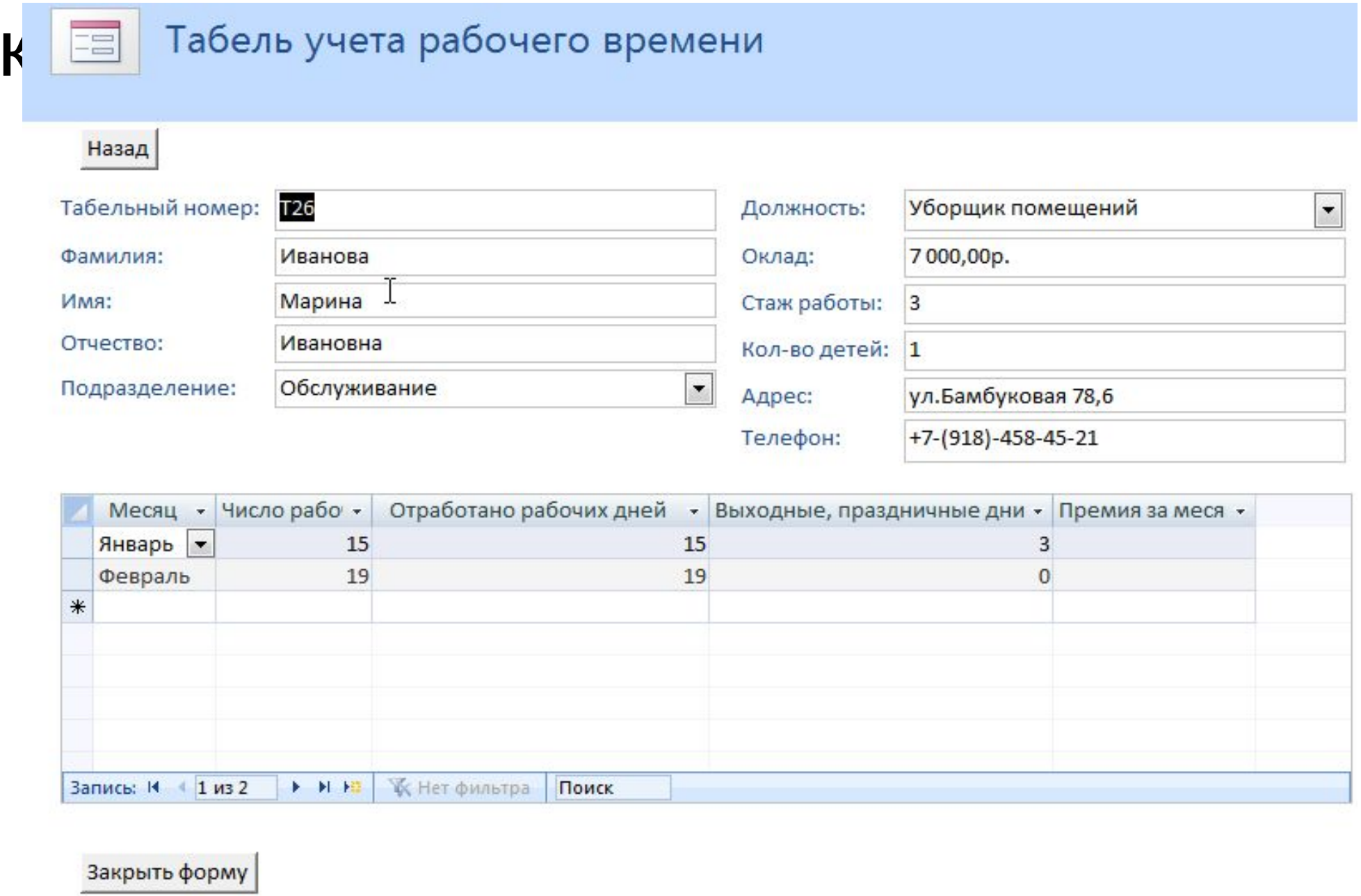
Сх


Подразделения
Код_подразделения
Название_подразделения
Должности
Код_должности
Название_должности
Месяц
Код_месяца
Название месяца
Рабочие дни
Праздничные дни
Выходные дни
Сотрудники
Табельный номер
Фамилия
Имя
Отчество
Подразделение
Должность
Оклад
Стаж работы
Кол-во детей
Адрес
Телефон
Ведомость
Табельный номер
Месяц
Число рабочих дней
Отработано рабочих дней
Выходные, праздничные дни
Премия за месяц
1
∞

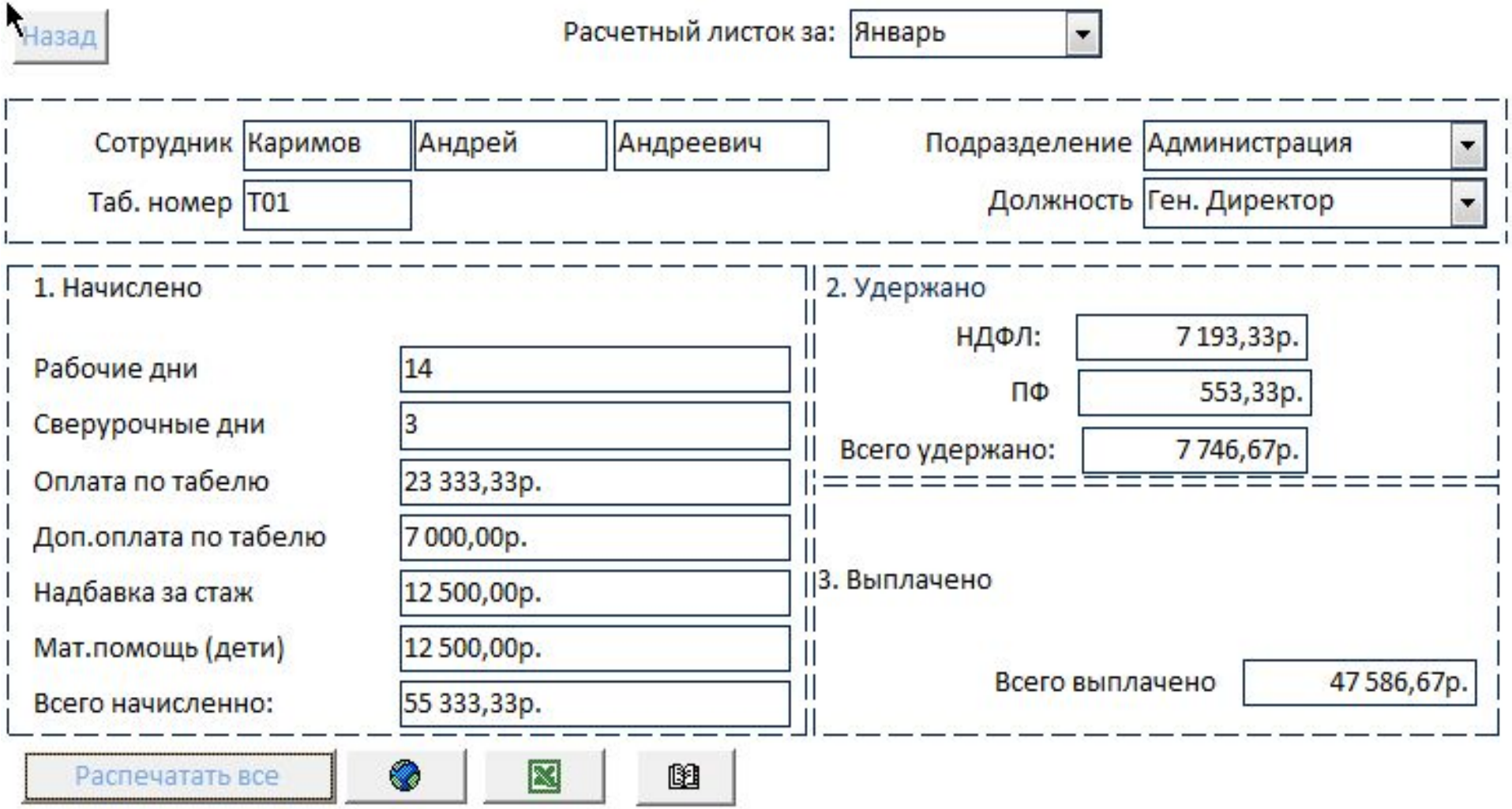
# Интерфейс системы
# Окно кнопочной формы



Ок


Табель учета рабочего времени
Назад
Табельный номер: Т26
Фамилия: Иванова
Имя: Марина
Отчество: Ивановна
Подразделение: Обслуживание
Должность: Уборщик помещений
Оклад: 7 000,00р.
Стаж работы: 3
Кол-во детей: 1
Адрес: ул.Бамбуковая 78,6
Телефон: +7-(918)-458-45-21
Месяц
Число рабо
Отработано рабочих дней
Выходные, праздничные дни
Премия за меся
Январь 15 15 3
Февраль 19 19 0
Запись: 1 из 2
Нет фильтра
Поиск
Закрыть форму

# Окно формы «Зарплата»



# Пример выходного документа

| Подразделение | Должность | Фамилия | Имя | Табельный номер | Отчество | |
|---|---|---|---|---|---|---|
| Администрация | Ген. Директор | Каримов | Андрей | T01 | Андреевич | 23 333,33р. |
| Администрация | Зам. ген. директора | Борисов | Сергей | T02 | Сергеевич | 20 000,00р. |
| Администрация | Секретарь | Тюрин | Иван | T03 | Николаевич | 10 000,00р. |
| Отдел кадров | Нач. отдела кадров | Соболева | Владислава | T04 | Николаевна | 15 600,00р. |
| Отдел кадров | Секретарь | Федоров | Денис | T05 | Альбертович | 10 000,00р. |
| Отдел кадров | Глав. Бухгалтер | Морозова | Евгения | T06 | Олеговна | 18 000,00р. |
| Бухгалтерия | Бухгалтер | Орлова | Ксения | T07 | Федоровна | 15 000,00р. |
| Бухгалтерия | Бухгалтер | Яковлев | Семен | T08 | Игоревич | 15 000,00р. |
| Произв. Отдел | Нач. производ. Отдела | Волков | Николай | T09 | Егорович | 19 000,00р. |
| Произв. Отдел | Архитектор | Лебедев | Семен | T10 | Семенович | 18 000,00р. |
| Произв. Отдел | Инженер-проектировщик | Павлов | Александр | T11 | Александрович | 17 000,00р. |
| Произв. Отдел | Инженер-программист | Кирилов | Кирилл | T12 | Алексеевич | 16 000,00р. |
| Произв. Отдел | Техник-программист | Яковлева | Анфиса | T13 | Леонидовна | 14 000,00р. |
| Произв. Отдел | Художник 3D графики | Русакова | Ульяна | T14 | Григорьевна | 15 000,00р. |
| Произв. Отдел | Художник 3D графики | Марков | Илья | T15 | Михайлович | 15 000,00р. |
| Произв. Отдел | Художник 3D графики | Медведева | Екатерина | T16 | Борисовна | 15 000,00р. |
| Произв. Отдел | Дизайнер | Панфилов | Михаил | T17 | Александрович | 13 000,00р. |

Изучена предметная область
Создано техническое задание
Разработана логическая и физическая модель БД
БД реализована в MS Access

# База данных «Ж/Д вокзал»

# Цель создания базы данных

- Повышение эффективности системы пассажирских Ж/Д перевозок на основе использования современных информационных технологий.

# Задачи решаемые Ж/Д вокзалом

- Управление Ж/Д потоком
- Предоставление информации о проходящих рейсах
- Продажа билетов на поезда

# Входные документы
# Расписание

| Номер поезда | Станция | Время прибытия | Остановка | Цена СВ | Цена Купе | Цена плацкарт |
|---|---|---|---|---|---|---|
| 343И | Челябинск | 1:30 | 0:00 | 0,00р. | 0,00р | 0,00р. |
| 343И | Оренбург | 2:40 | 0:05 | 280,00р. | 210,00р. | 150,00р. |
| 343И | Самара | 2:55 | 0:03 | 460,00р. | 290,00р. | 235,00р. |
| 343И | Сызрань | 3:20 | 0:10 | 550,00р. | 335,00р. | 270,00р. |
| 343И | Саратов | 4:40 | 0:02 | 700,00р. | 420,00р. | 330,00р. |
| 343И | Волгоград | 5:10 | 0:03 | 900,00р. | 490,00р. | 360,00р. |
| 343И | Ея | 5:40 | 0:04 | 1 460,00р. | 850,00р. | 620,00р. |
| 343И | Тихорецкая | 6:50 | 0:12 | 1 600,00р. | 945,00р. | 790,00р. |
| 343И | Выселки | 8:20 | 0:08 | 1 655,00р. | 980,00р. | 820,00р. |
| 343И | Кореновск | 9:33 | 0:02 | 1 690,00р. | 1 000,00р. | 835,00р. |
| 343И | Краснодар | 10:40 | 0:07 | 1 825,00р. | 1 060,00р. | 870,00р. |
| 343И | Горячий ключ | 12:20 | 0:35 | 1 900,00р. | 1 095,00р. | 890,00р. |
| 343И | Туапсе | 14:10 | 0:11 | 1 955,00р. | 1 135,00р. | 915,00р. |
| 343И | Сочи | 16:32 | 0:30 | 2 085,00р. | 1 325,00р. | 960,00р. |
| 343И | Адлер | 18:20 | 0:00 | 2 150,00р. | 1 370,00р. | 980,00р. |

# Выходной документ

Билет № 29

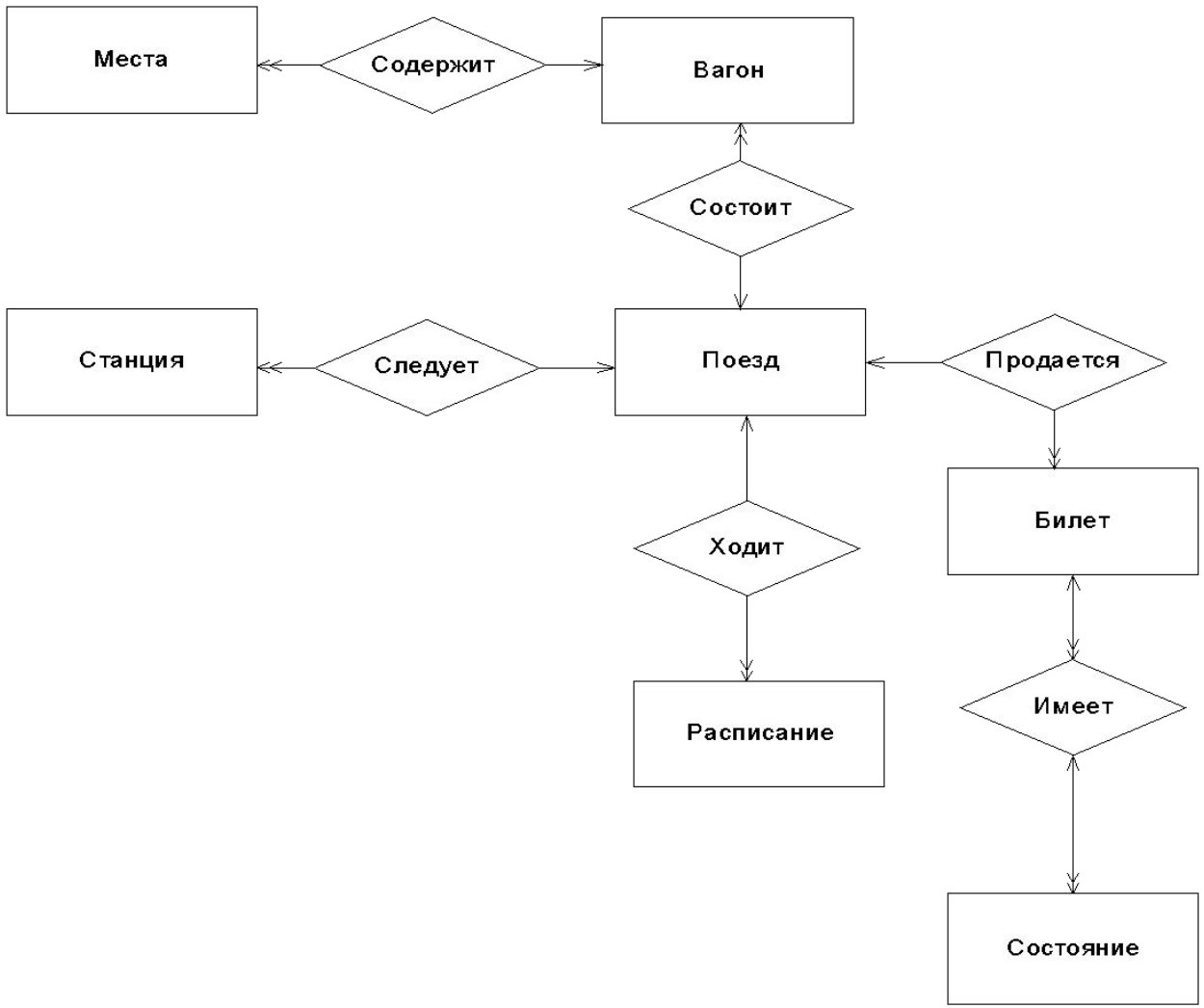
| Номер поезда | Станция отправления | Станция прибытия |
|---|---|---|
| 353Е | Сызрань | Кореновск |

Цена 1 615,00р.

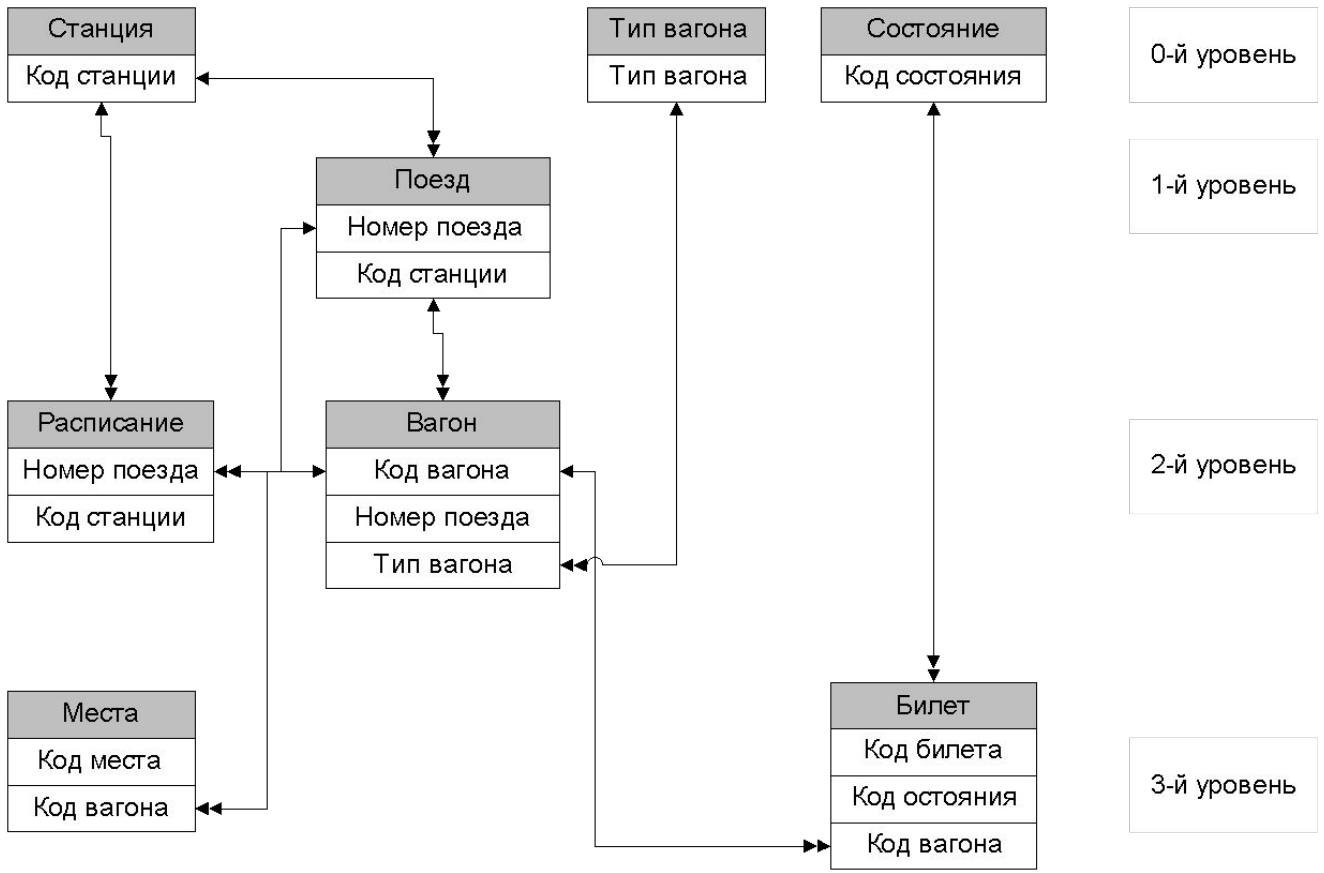
| | | | | | |
|---|---|---|---|---|---|
| Номер паспорта | 5265 734767 | Дата отправления и приытия | 14.01.2011 | Номер вагона | 3 |
| Фамилия Имя | Бортников КТ | Время отправления | 14:15 | Тип вагона | купе |
| | | Время прибытия | 19:50 | Номер места | 28 |

Дата покупки 10.01.2011

# ER –диаграмма (диаграмма сущность – связь)



# Логическая схема базы данных



# Пример описания физической модели Представим физическую модель в виде таблиц

### Станция

| Имя поля | Ключевое поле | Тип данных | Размер поля |
|---|---|---|---|
| Код_станции | Да | Текстовый | 6 |
| Название | Нет | Текстовый | 50 |

### Места

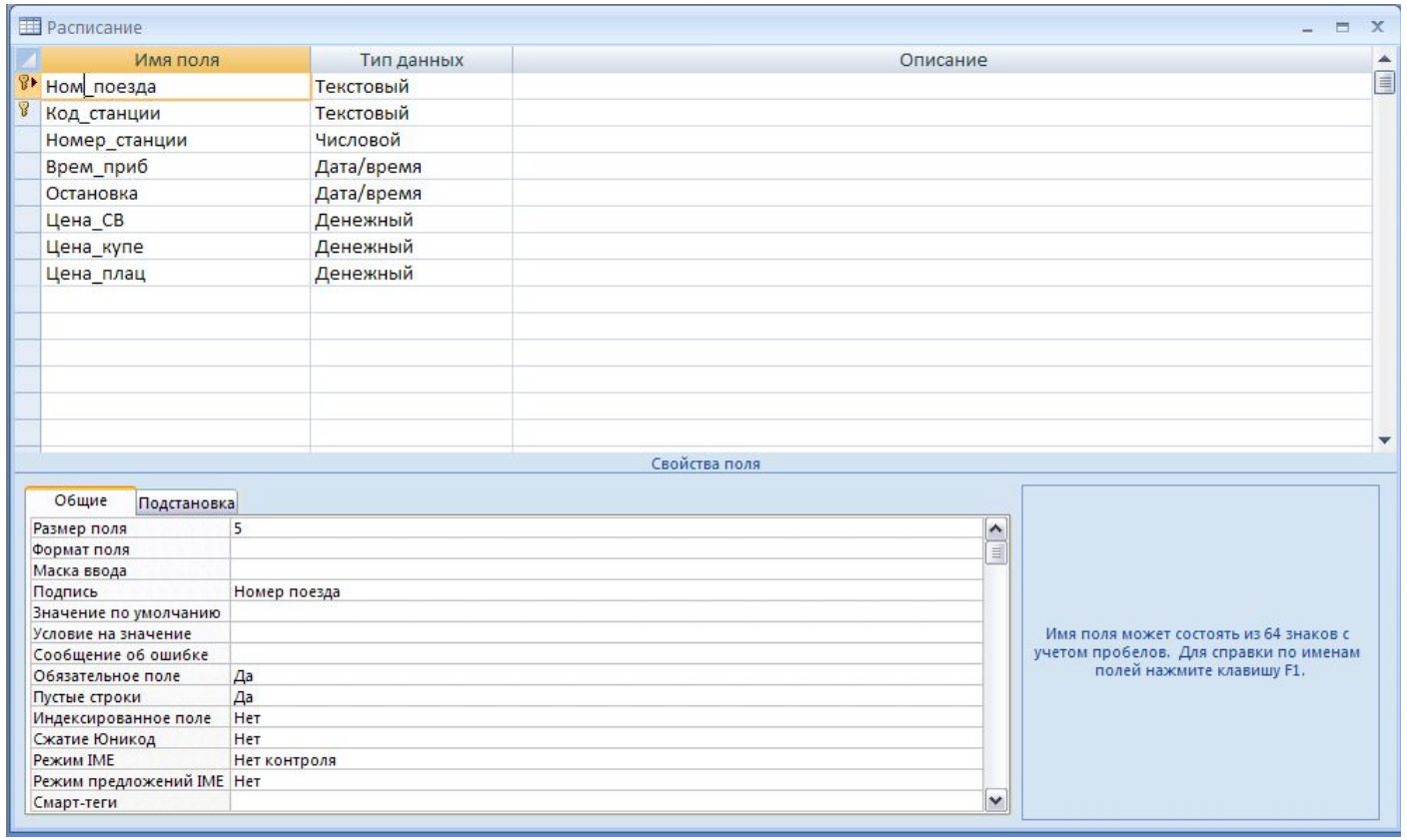
| Имя поля | Ключевое поле | Тип данных | Размер поля | Формат |
|---|---|---|---|---|
| Код_места | Да | Счетчик | | Длинное целое |
| Код_вагона | Нет | Текстовый | 5 | |
| Номер_места | Нет | Числовой | | Байт |

### Поезд

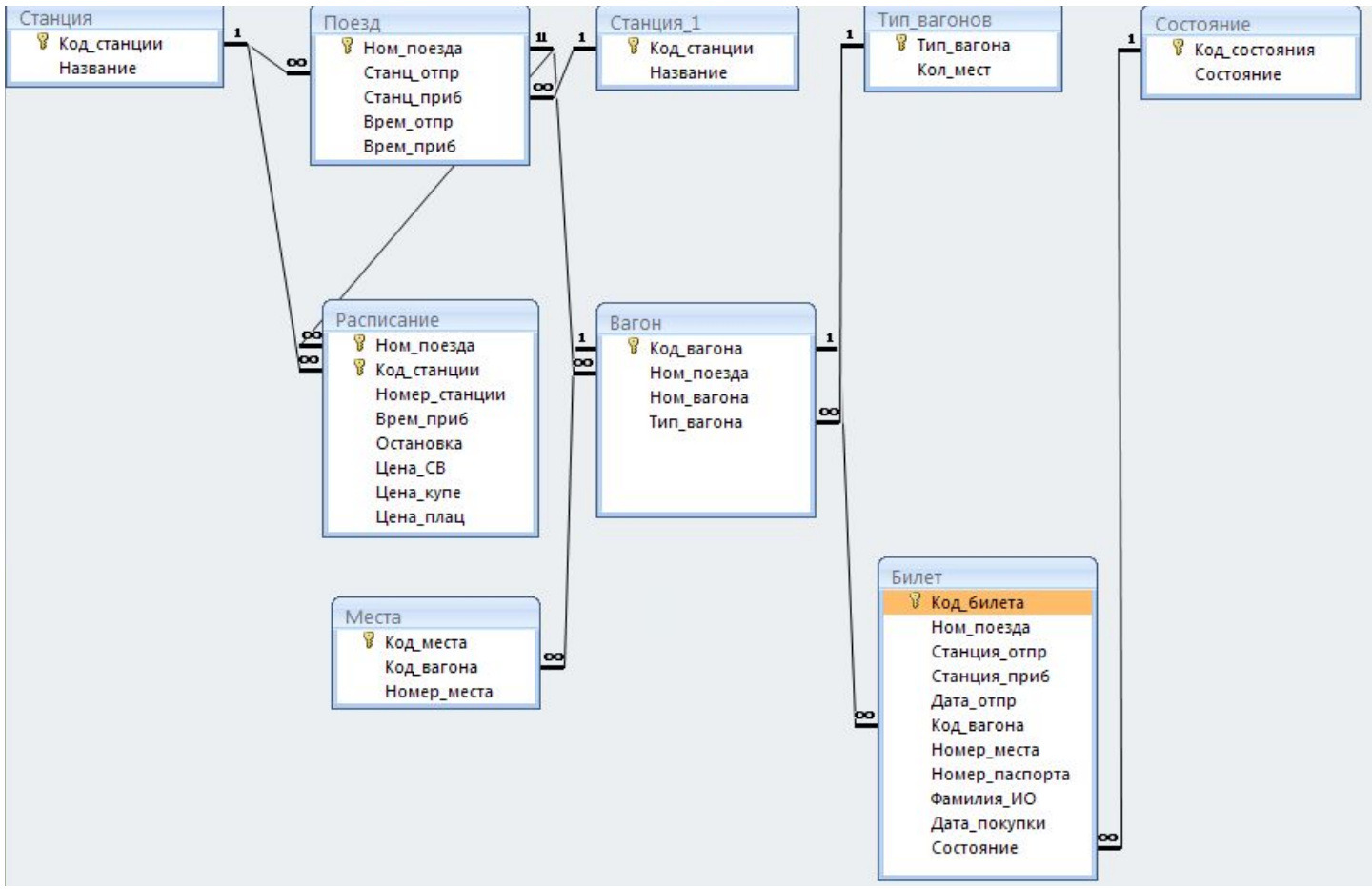
| Имя поля | Ключевое поле | Тип данных | Размер поля | Формат |
|---|---|---|---|---|
| Ном_поезда | Да | Текстовый | 5 | |
| Станция_отпр | Нет | Текстовый | 6 | |
| Станция_приб | Нет | Текстовый | 6 | |
| Время_отпр | Нет | Дата/время | | Краткий формат времени |
| Время_приб | Нет | Дата/время | | Краткий формат времени |

# Реализация базы данных
# Создание таблиц

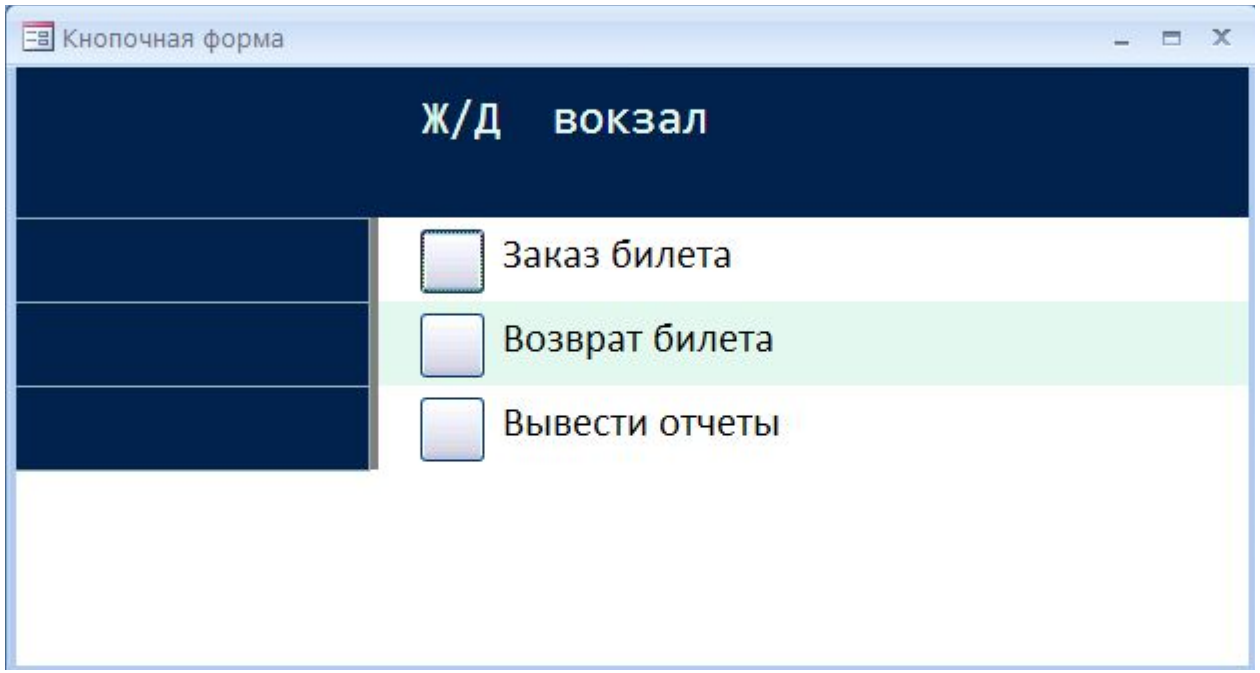
**Для создания таблиц воспользуемся конструктором, создадим 8 таблиц в соответствии с физической моделью БД**



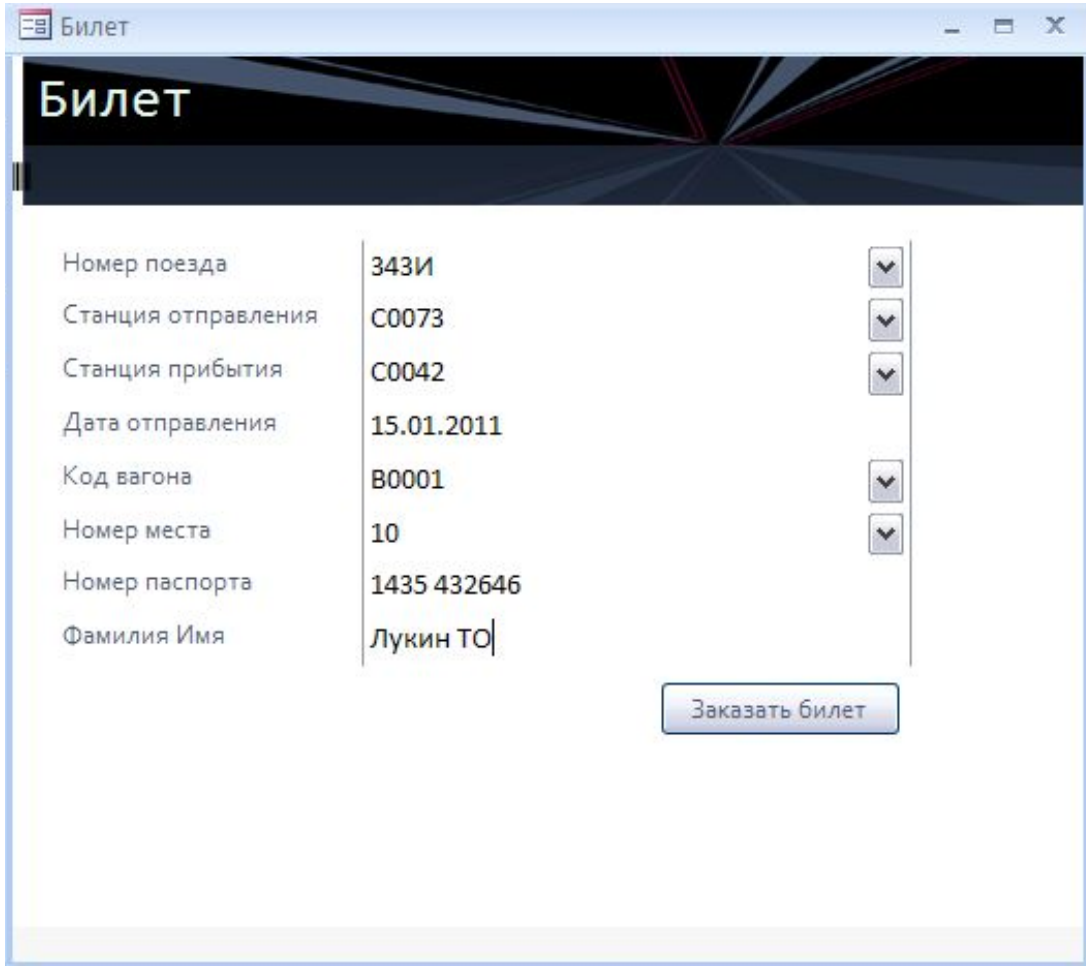
# Схема данных



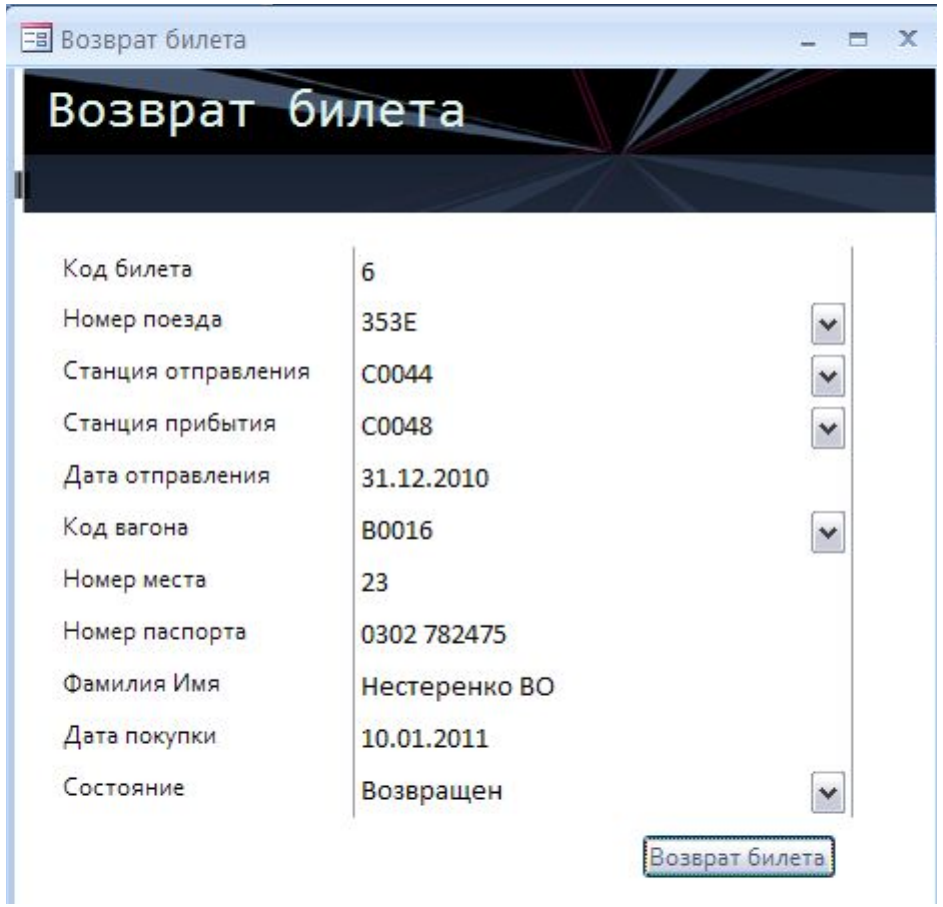
# Интерфейс системы
# Окно кнопочной формы



# Форма заказа билета



# Возврат билета осуществляется через окно возврат билета



# Вывод отчетов