**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Механико-математический факультет**

**Техническое обслуживание станков**

(Вариант 13)

Дрозд Милены, Рубан Александры, студентов 2 курса 9 группы

Минск, 2023

Оглавление

[1 Проектирование базы данных «Техническое обслуживание станков» 3](#_Toc130491981)

[1.1 Постановка задачи 3](#_Toc130491982)

[1.2 Инициирование проекта (фаза 0) 3](#_Toc130491983)

[1.3 Определение множеств сущностей (фаза 1) 3](#_Toc130491984)

[1.3.1 Выделение множеств сущностей 3](#_Toc130491985)

[1.3.2 Описание множеств сущностей 4](#_Toc130491986)

[1.4 Построение модели уровня сущностей (фаза 2) 4](#_Toc130491987)

[1.4.1 Матрица связей 4](#_Toc130491988)

[1.4.2 Описание связей 4](#_Toc130491989)

[1.4.3 Концептуальная схема уровня сущностей 5](#_Toc130491990)

[1.5 ПОСТРОЕНИЕ МОДЕЛИ УРОВНЯ КЛЮЧЕЙ (ФАЗА 3) 5](#_Toc130491991)

[1.5.1 Разрешение неопределённых связей 5](#_Toc130491992)

[1.5.2 Определение ключевых атрибутов и доменов 6](#_Toc130491993)

[1.5.3 Концептуальная схема 8](#_Toc130491994)

[1.6 ПОСТРОЕНИЕ ПОЛНОАТРИБУТНОЙ МОДЕЛИ (ФАЗА 4) 8](#_Toc130491995)

[1.6.1 Явные ограничения целостности 9](#_Toc130491996)

[1.7 ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ 9](#_Toc130491997)

[1.8 Запросы на проверку ограничений 14](#_Toc130491998)

# Проектирование базы данных «Техническое обслуживание станков»

## Постановка задачи

Необходимо разработать систему отслеживания финансовой стороны деятельности предприятия в соответствии со следующим описанием.

* Предприятие занимается ремонтом станков и другого промышленного оборудования.
* Клиентами компании являются промышленные предприятия, оснащенные различным сложным оборудованием.
* Каждое промышленное оборудование нуждается в ремонте.
* Ремонтные работы классифицируются по видам.

## Инициирование проекта (фаза 0)

Допустим, что на этой фазе были определены все участники проекта, определены их функции. Уточним цели проектирования.

В результате предварительного изучения поставленной задачи определены требования к разрабатываемой системе:

* Необходимо отобразить в системе состояние отслеживания финансовой стороны деятельности данного предприятия.
* Клиенты обращаются к данному предприятию в случае поломок оборудования.
* Ремонтные работы в компании организованы следующим образом: все станки проклассифицированы по странам-производителям, годам выпуска и маркам.
* Все виды ремонта отличаются названием, продолжительностью в днях, стоимостью.
* По каждому факту ремонта фиксируется вид станка и дата начала ремонта.

## Определение множеств сущностей (фаза 1)

### Выделение множеств сущностей

В соответствии с описанием предметной области можно выделить следующие множества сущностей:

*Станки.*

*Вид ремонта.*

### Описание множеств сущностей

Описание выделенных множеств сущностей приведено в табл. 1.

Таблица 1. Пул сущностей

|  |  |  |
| --- | --- | --- |
| **Имя множества сущностей** | **Определение множества сущностей** | **Описание множества сущностей** |
| Станки | Техническое устройство, выполняющее механические движения для преобразования энергии, материалов и информации. | Промышленное оборудование поступает на предприятие от клиента.  Как только от клиента поступает запрос на ремонт промышленного оборудования, формируется новый экземпляр данного множества сущностей. |
| Вид ремонта | Комплекс мероприятий по восстановлению работоспособного или исправного состояния какого-либо объекта | Как только конкретное промышленное оборудование, нуждающееся в восстановлении, поступает на предприятие, формируется новый экземпляр данного множества сущностей. |

Полезно привести примеры выделенных сущностей:

Станки: токарный, расточный, сверлильный.

Вид ремонта: косметический, восстановительный, текущий.

## Построение модели уровня сущностей (фаза 2)

### Матрица связей

На основании анализа постановки задачи устанавливаются связи между множествами сущностей. Матрице связей приведена в табл. 2.

Таблица 2. Матрица связей

|  |  |  |
| --- | --- | --- |
|  | **Станки** | **Вид ремонта** |
| **Станки** |  | x |
| **Вид ремонта** | x |  |

### Описание связей

Описание выделенных связей приводится в табл. 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1-я сущность** | **2-я сущност**ь | **Имя связи** | **Тип связи** | **Мощность связи** | **Описание связи** |
| Станки | Вид ремонта | нуждается в/предоставляется для | неопределённая | P; B | Каждый станок нуждается в *одном или более* видах ремонта. Каждая ремонтная работа предоставляется для *одного или более* видов станков. |

Таблица 3. Описание связей

### Концептуальная схема уровня сущностей

В соответствии с определенными множествами сущностей и связями между ними строится концептуальная схема уровня сущностей ([рис. 1](#_bookmark0)).

нуждается в/

предоставляется для

Станки

Вид ремонта

Рис. 1. Концептуальная схема уровня сущностей

## ПОСТРОЕНИЕ МОДЕЛИ УРОВНЯ КЛЮЧЕЙ (ФАЗА 3)

### Разрешение неопределённых связей

Выполним разрешение неопределённой связи (типа "многие ко многим"). Введём дополнительную сущность, описание которой включим в пул сущностей (табл. 4). Соответственно, внесём необходимые изменения в определение связей: добавим новые связи, уточним типы связей (табл. 5).

Таблица 4. Пул сущностей

|  |  |  |
| --- | --- | --- |
| **Имя множества сущностей** | **Определение множества сущностей** | **Описание множества сущностей** |
| Станки | Техническое устройство, выполняющее механические движения для преобразования энергии, материалов и информации. | Промышленное оборудование поступает на предприятие от клиента.  Как только от клиента поступает запрос на ремонт промышленного оборудования, формируется новый экземпляр данного множества сущностей. |
| Ремонт | Информация о ремонтной работе для определённого станка. | Как только какому-либо промышленному оборудованию назначается вид ремонта, формируется новый экземпляр данного множества сущностей. |
| Вид ремонта | Комплекс мероприятий по восстановлению работоспособного или исправного состояния какого-либо объекта | Как только конкретное промышленное оборудование, нуждающееся в восстановлении, поступает на предприятие, формируется новый экземпляр данного множества сущностей. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1-я сущность** | **2-я сущност**ь | **Имя связи** | **Тип связи** | **Мощность связи** | **Описание связи** |
| Станки | Ремонт | нуждается в | идент. опред. | P | Каждый станок нуждается в *одной или более* ремонтных работах. Каждая ремонтная работа предоставляется *только одному* станку. |
| Вид ремонта | Ремонт | принадлежит | идент. опред. | Р | Каждый вид ремонта принадлежит *одной или более* ремонтной работе. Каждая ремонтная работа включает *только один* вид ремонта. |

Таблица 5. Определение связей

### Определение ключевых атрибутов и доменов

Определение ключевых атрибутов осуществляется одновременно с определением доменов, на которых определяются эти атрибуты. Сначала из анализа задачи для каждого множества сущностей выделяются ключи (простые или составные), позволяющие идентифицировать экземпляры сущностей. В случае если ключей несколько, один выбирается в качестве первичного ключа, а остальные определяются как альтернативные. Связи между множествами сущностей представляются с помощью внешних ключей. Таким образом, определяются все ключи. Для каждого атрибута определяется домен.

В рассматриваемом примере для выделенных сущностей можно предложить следующие ключи и домены:

**СТАНКИ**

Первичный ключ: Код станка. Значениями атрибута являются натуральные числа (порядковый номер в перечислении станков), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

**ВИД РЕМОНТА**

Первичный ключ: Код вида. Значениями атрибута являются натуральные числа (порядковый номер в перечислении видов ремонта), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

**РЕМОНТ**

Первичный ключ: Id. Значениями атрибута являются натуральные числа (порядковый номер в перечислении ремонта), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

Внешние ключи:

Код станка – устанавливает связь с множеством сущностей СТАНКИ;

Атрибут определён на типе данных int;

Код вида – устанавливает связь с множеством сущностей ВИД РЕМОНТА; атрибут определён на типе данных int;

Описание ключевых атрибутов приведено в табл. 6.

Таблица 6. Описание атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Множество сущностей** | **Имя атрибута** | **Тип данных** | **Признак обязательности** | **Примечание** |
| Станки | Код станка | int | Not null | PK |
| Ремонт | Id | int | Not null | PK |
| Код станка | int | Not null | FK1 |
| Код вида | int | Not null | FK2 |
| Вид ремонта | Код вида | int | Not null | PK |

### Концептуальная схема

По результатам определения первичных ключей строится схема уровня ключей ([рис. 2](#_bookmark1)).



Рис. 2. Концептуальная схема уровня ключей

## ПОСТРОЕНИЕ ПОЛНОАТРИБУТНОЙ МОДЕЛИ (ФАЗА 4)

На этом этапе в соответствии с проведенным анализом предметной области определяются все необходимые атрибуты и домены, на которых эти атрибуты определены. Соответствующим образом модифицируются таблица описания доменов и атрибутов (табл. 7) и строится полноатрибутная концептуальная схема ([рис. 4](#_bookmark2)).

Таблица 7. Описание атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Множество сущностей** | **Имя атрибута** | **Тип данных** | **Признак обязательности** | **Примечание** |
| Станки | Код станка | Int, identity | Not null | PK |
| Производитель | Nvarchar (30) | Not null | Страна-производитель |
| Год выпуска | Int | Not null |  |
| Марка | Nvarchar (30) | Not null |  |
| Ремонт | Id | Int | Not null | PK |
| Код станка | Int | Not null | FK1 |
| Код вида | Int | Not null | FK2 |
| Вид станка | Nvarchar (30) | Not null |  |
| Дата начала | Date | Not null |  |
| Вид ремонта | Код вида | Int, identity | Not null | PK |
| Название | Nvarchar (30) | Not null |  |
| Продолжительность | Int | Not null | Продолжительность в днях |
| Стоимость | Money | Not null |  |

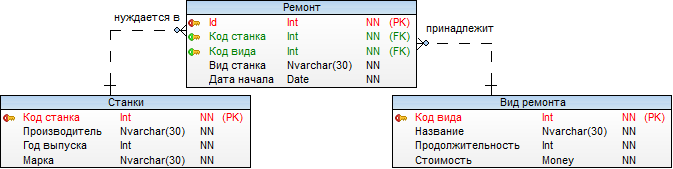


Рис. 3. Полноатрибутная концептуальная схема

### Явные ограничения целостности

Из анализа поставленной задачи можно выделить следующие дополнительные ограничения целостности:

все атрибуты с типом данных Int должны иметь исключительно положительные значения.

## ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ

Создадим базу данных ТО Станков / MachinesService.

USE master;

GO

DROP DATABASE IF EXISTS MachinesService;

GO

CREATE DATABASE MachinesService;

GO

USE MachinesService;

GO

В соответствии с приведенной структурой, внутренняя схема разрабатываемой базы данных представлена в табл. 8 –15.

Таблица 8. Станки / Machines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код станка | MachineCode | Int, identity | Not null | PK |
| Производитель | CountryCreator | Nvarchar (30) | Not null |  |
| Год выпуска | YearOfRelease | Int | Not null |  |
| Марка | Mark | Nvarchar (30) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Machines]

(

[MachineCode] Int IDENTITY NOT NULL PRIMARY KEY CLUSTERED,

[CountryCreator] Nvarchar(30) NOT NULL,

[YearOfRelease] Int NOT NULL,

[Mark] Nvarchar(30) NOT NULL,

CONSTRAINT CK\_Machines\_Year CHECK (YearOfRelease > 0)

)

GO

Таблица 9. Вид ремонта / RepairType

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код вида | TypeCode | Int, identity | Not null | PK |
| Название | Name | Nvarchar (30) | Not null |  |
| Продолжительность | Duration | Int | Not null |  |
| Стоимость | Price | Money | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [RepairType]

(

[TypeCode] Int IDENTITY NOT NULL PRIMARY KEY CLUSTERED,

[Name] Nvarchar(30) NOT NULL,

[Duration] Int NOT NULL,

[Price] Money NOT NULL,

CONSTRAINT CK\_Type\_Duration CHECK (Duration > 0)

)

GO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Id | Id | Int | Not null | PK |
| Код станка | MachineCode | Int | Not null | FK1 |
| Код вида | TypeCode | Int | Not null | FK2 |
| Вид станка | MachineType | Nvarchar (30) | Not null |  |
| Дата начала | DateOfStart | Date | Not null |  |

Таблица 10. Ремонт / Repair

Описание таблицы на SQL:

CREATE TABLE [Repair]

(

[Id] Int IDENTITY NOT NULL PRIMARY KEY CLUSTERED,

[MachineCode] Int NOT NULL,

CONSTRAINT FK\_Repair\_Machine FOREIGN KEY (MachineCode)

REFERENCES Machines(MachineCode),

[TypeCode] Int NOT NULL,

CONSTRAINT FK\_Repair\_Type FOREIGN KEY (TypeCode)

REFERENCES RepairType(TypeCode),

[MachineType] Nvarchar(30) NOT NULL,

[DateOfStart] Date NOT NULL,

)

GO

Заполняем таблицу станков (Machines):

INSERT INTO Machines (CountryCreator, YearOfRelease, Mark)

VALUES

('USA', 2015, 'Caterpillar'),

('Japan', 2010, 'Komatsu'),

('Germany', 2018, 'Liebherr'),

('USA', 2020, 'John Deere'),

('Japan', 2017, 'Hitachi'),

('USA', 2019, 'Bobcat'),

('China', 2016, 'XCMG')

GO

SELECT \*

FROM Machines

GO

Заполняем таблицу видов ремонта (RepairType):

INSERT INTO RepairType ([Name], Duration, Price)

VALUES

('Regular Maintenance', 8, 100.00),

('Oil Change', 2, 50.00),

('Hydraulic System Repair', 20, 500.00),

('Electrical System Repair', 16, 400.00),

('Engine Overhaul', 40, 1000.00),

('Transmission Repair', 24, 600.00),

('Brake System Repair', 12, 300.00),

('Suspension Repair', 18, 450.00),

('Tire Replacement', 4, 200.00),

('Body Repair', 30, 750.00);

SELECT \*

FROM RepairType

GO

Заполняем таблицу ремонта (Repair):

INSERT INTO Repair (MachineCode, TypeCode, MachineType, DateOfStart)

VALUES

(1, 1, 'Bulldozer', '2022-01-05'),

(2, 3, 'Excavator', '2022-02-10'),

(3, 5, 'Crane', '2022-03-15'),

(4, 7, 'Backhoe', '2022-04-20'),

(5, 9, 'Loader', '2022-05-25'),

(6, 2, 'Forklift', '2022-06-30');

GO

SELECT \*

FROM Repair

GO

Более понятный запрос для ремонта по станкам:

SELECT R.MachineCode, R.TypeCode, R.MachineType, R.DateOfStart

FROM Repair AS R

INNER JOIN Machines AS M ON R.MachineCode = M.MachineCode;

GO

Более понятный запрос для ремонта по видам ремонта:

SELECT T.Duration, T.Name, T.Price, T.TypeCode

FROM RepairType AS T

INNER JOIN Repair AS R ON R.TypeCode = T.TypeCode;

GO

Продемонстрируем случай возникновения ошибки при вставке строк в таблицу Machines.

Конфликт инструкции INSERT с ограничением CHECK “CK\_Machines\_Year”:

INSERT INTO Machines (CountryCreator, YearOfRelease, Mark)

VALUES

('USA', 2015, 'Caterpillar'),

('Japan', 2010, 'Komatsu'),

('Germany', -2018, 'Liebherr');

--Атрибут YearOfRelease должен быть исключительно

--положительным, однако в выше написанных строчках наблюдается --отрицательное значение

GO

Успешная вставка:

INSERT INTO Machines (CountryCreator, YearOfRelease, Mark)

VALUES

('USA', 2015, 'Caterpillar'),

('Japan', 2010, 'Komatsu'),

('Germany', 2018, 'Liebherr');

GO

Продемонстрируем случай возникновения ошибки при вставке строк в таблицу RepairType.

Конфликт инструкции INSERT с ограничением CHECK “CK\_Type\_Duration”:

INSERT INTO RepairType ([Name], Duration, Price)

VALUES

('Regular Maintenance', 8, 100.00),

('Oil Change', 2, 50.00),

('Hydraulic System Repair', -20, 500.00)

--Атрибут Duration должен быть исключительно

--положительным, однако в выше написанных строчках наблюдается --отрицательное значение

GO

Успешная вставка:

INSERT INTO RepairType ([Name], Duration, Price)

VALUES

('Regular Maintenance', 8, 100.00),

('Oil Change', 2, 50.00),

('Hydraulic System Repair', 20, 500.00)

GO

## Запросы на проверку ограничений

Запросы на проверку ограничений:

* Проверка ограничения CHECK на поле YearOfRelease в таблице Machines:

SELECT \* FROM Machines WHERE YearOfRelease <= 0;

Если в результате выполнения этого запроса будут найдены записи, значит, ограничение не соблюдается.

* Проверка ограничения CHECK на поле Duration в таблице RepairType:

SELECT \* FROM RepairType WHERE Duration <= 0;

Если в результате выполнения этого запроса будут найдены записи, значит, ограничение не соблюдается.

* Проверка наличия внешних ключей в таблице Repair:

SELECT \*

FROM Repair

WHERE MachineCode NOT IN (SELECT MachineCode FROM Machines)

OR TypeCode NOT IN (SELECT TypeCode FROM RepairType);

Если в результате выполнения этого запроса будут найдены записи, значит, есть записи в таблице Repair, которые ссылается на несуществующие записи в таблицах Machines или RepairType.

* Проверка уникальности комбинаций значений полей MachineCode и TypeCode в таблице Repair:

SELECT MachineCode, TypeCode, COUNT(\*)

FROM Repair

GROUP BY MachineCode, TypeCode

HAVING COUNT(\*) > 1;

Если в результате выполнения этого запроса будут найдены записи, значит, в таблице Repair есть дубликаты записей с одинаковыми значениями полей MachineCode и TypeCode.