НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсової роботи**

з дисципліни

«БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ»

на тему

«ІНФОРМАЦІЙНА СИСТЕМА АВІАБУДІВНОГО ПІДПРИЄМСТВА»

Викладач М. В. Коломицев

Студентка Поліщук О. С.

Група ФБ-11

Варіант 19

Київ – 2022

***Варіант 19 .***

**Інформаційна система авіабудівного підприємства**

Структурно підприємство розбите на цехи, які в свою чергу поділяються на ділянки.

Вироби, що випускаються підприємства - літаки (цивільні, транспортні, військові), планери, вертольоти, дельтаплани , ракети (артилерійські, авіаційні, військово-морські), інші вироби. Кожна категорія виробів має специфічні, притаманні лише їй атрибути. Наприклад, для літаків це число двигунів, для ракети - потужність заряду і т.д. По кожній категорії виробів може збиратися кілька видів виробів.

Кожній категорії інженерно-технічного персоналу (інженери, технологи, техніки) і робочих (збирачі, токарі, слюсарі, зварювальники та ін.) Також властиві характерні тільки для цієї групи атрибути. Робочі об'єднується в бригади, якими керують бригадири. Бригадири обираються з числа робітників, майстри, начальники ділянок і цехів призначаються з числа інженерно-технічного персоналу.

Кожен виріб збирається в своєму цеху (в цеху може збиратися кілька видів виробів) і в процесі виготовлення проходить певний цикл робіт, переміщаючись з однієї ділянки на іншу. Всі роботи по збірці конкретного виробу на певній ділянці виконує одна бригада робітників, при цьому на ділянці може працювати кілька бригад. Очолює роботу на ділянці начальник ділянки, в підпорядкуванні якого знаходиться кілька майстрів. Різні вироби можуть проходити одні й ті ж цикли робіт на одних і тих же ділянках цеху.

Ведеться облік руху кадрів і облік продукції, що випускається.

Види запитів в інформаційній системі:

1. Отримати перелік видів виробів окремої категорії і в цілому, що збираються зазначеним цехом, підприємством.

2. Отримати число і перелік виробів окремої категорії і в цілому, зібраних зазначеним цехом, ділянкою, підприємством в цілому за певний відрізок часу.

3. Отримати дані про кадровий склад цеху, підприємства в цілому і по зазначеним категоріям інженерно-технічного персоналу і робітників.

4. Отримати число і перелік ділянок зазначеного цеху, підприємства в цілому і їх начальників.

5. Отримати перелік робіт, які проходить вказаний виріб.

6. Отримати склад бригад вказаної ділянки, цеху.

7. Отримати список майстрів вказаної ділянки, цеху.

8. Отримати перелік виробів окремої категорії і в цілому, що збираються зараз зазначеною ділянкою, цехом, підприємством.

9. Отримати склад бригад, що беруть участь в складанні зазначеного виробу.

Передбачити розробку тригерів, які забезпечують каскадні зміни в зв'язаних таблицях.

**ЗМІСТ**

[1. Вступ 5](#_Toc122617364)

[2. Аналіз предметної області 8](#_Toc122617365)

[3. Функціональне моделювання предметної області з використанням DFD-діаграми 10](#_Toc122617366)

[4. Концептуальна модель бази даних 12](#_Toc122617367)

[4.1 Вибір сутностей 12](#_Toc122617368)

[4.2 Визначення атрибутів 13](#_Toc122617369)

[4.3 Визначення зв’язків 17](#_Toc122617370)

[4.4 Результати концептуального моделювання 19](#_Toc122617371)

[5. Реляційна модель бази даних 20](#_Toc122617372)

[6. Фізична модель бази даних 21](#_Toc122617373)

[6.1 Створення таблиць 21](#_Toc122617374)

[6.2 Забезпечення цілісності бази даних 27](#_Toc122617375)

[6.3 Приклад заповненої бази даних 31](#_Toc122617376)

[7. Реалізація завдань 36](#_Toc122617377)

[8. Висновки по роботі 48](#_Toc122617378)

[Список використаних джерел 49](#_Toc122617379)

1. **Вступ**

ТУТ БУВ ВСТУП НА 2 СТОРІНКИ, А ПОТІМ Я ТАКА

*Метою роботи* є ОЦЕ

*Завданням дослідження* є ОЦЕ

*Порядком виконання поставленого дослідження* є ОЦЕ

Задля виконання дослідження поставленої предметної області використовуються засоби реалізації CASE Studio 2 для побудови ER-моделі та DFD-діаграми, MS SQL Server задля створення спроектованої бази даних.

1. **Аналіз предметної області**

За допомогою аналізу даної ПО, можна сформувати єдиний погляд на дані відомості та інформацію, що обробляється. Треба враховувати усі правила даної СУБД, що будуть відображатися у виділених запитах.

Структура авіабудівного підприємства передбачає собою цехи по виготовленню конкретних авіа-виробів, які відрізняються між собою характеристиками використання. Цехи поділяються на ділянки, що в свою чергу передбачає конкретний цикл роботи, яку виконують робочі. Вони об’єднуються у бригади, котрими керують бригадири. За основні процеси на ділянках відповідають майстри у підпорядкуванні начальника ділянки, а за цех відповідає начальник цеху. Останні є частиною технічного персоналу підприємства.

**Вхідні дані:**

* Відомості про вироби: код, тип, назва, категорія, характеристики;
* Відомості про робочих: П.І.Б., дата народження, спеціалізація, посада, зарплатня, бригада;
* Відомості про технічний персонал: П.І.Б., дата народження, спеціалізація, посада, зарплатня, ділянка або цех;
* Цехи: Назва, начальник цеху, номер приміщення;
* Ділянки: Вид роботи, начальник ділянки, цех.
* Відомості про виготовлення виробів: ділянка, бригада, виріб, дата початку роботи, дата кінця роботи.

**Вихідні дані:**

* Облік виготовлення виробів;
* Формування списків працівників;
* Облік утворених бригад та груп майстрів;
* Виведення структури підприємства;

**Функціонал:**

* Ведення обліку виготовлення товарів зі зазначенням робочих та циклу роботи;
* Ведення обліку кадрового складу підприємства та їх роботи;
* Ведення обліку товарів та цеху, ділянці, в яких вони виготовляються;
* Забезпечення перегляду усієї структури підприємства;

1. **Функціональне моделювання предметної області з використанням DFD-діаграми**

ТУТ Я РОСКАЗАЛА ЩО ТАКЕ ДФД ДІГРАМА, ЇЇ ВИКОРИСТАННЯ І ТД

*Зовнішні об’єкти:* Начальник цеху, Начальники ділянок, Бригади;

*Сховища інформації:* Вироби, Каталог категорій виробів, Технічний персонал, Робочі, Стан замовлення;

*Процеси:* Замовлення на виріб, Розподіл процесів роботи між ділянками, Призначення робочих на ділянку, Встановлення термінів роботи на ділянці, Початок роботи, Закінчення роботи;

*Потоки даних:* Отримує замовлення, Каталог виробів-Інформація о виробі, Отримують інформацію щодо роботи, Розподіл роботи, Запит на інформацію-Отримують інформацію, Формування груп майстрів, Робота на ділянці, Формування бригад, Розподіл циклу роботи між робочими, Розпорядження, Приступають до роботи, Внесення змін щодо роботи, Оновлення інформації, Сповіщення о виготовленому виробі.

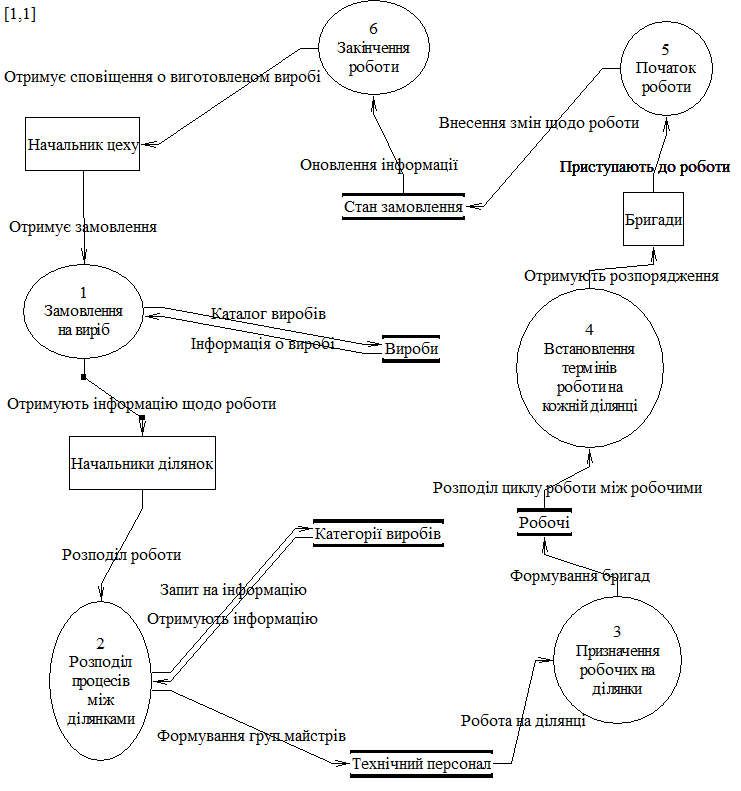


Рисунок 3.1 Data Flow Diagram

1. **Концептуальна модель бази даних**

ЩО ТРЕБА ЗРОБИТИ ЩОБ ПОБУДУВАТИ КОНЦЕПТУАЛЬН МОДЕЛЬ ТА ШО ВОНО ТАКЕ

**4.1 Вибір сутностей**

Таблиця сутностей 4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Найменування | Умовне позна-чення | Пер-винний ключ | Перелік завдань та запитів | Обмеження на доступність |
| 1 | Цехи | WORKSHOP | ID  цеху | Структура підприємства та виготовлення товарів | Загальний |
| 2 | Ділянки | DEPARTMENT | ID  ділянки | Структура підприємства, цикли роботи на виробом | Загальний |
| 3 | Вироби | PRODUCTS | ID  вироба | Назви виробів, що виготовляють | Загальний |
| 4 | Категорії виробів | CATEGORIES OF 'PRODUCT' | Номер категорії | Облік категорій виробів та характеристик | Загальний |
| 5 | Виготовлення виробів | PROGRESS | ID  роботи | Процес виготовлення виробів та облік роботи | Обмежений , слабка сутність |
| 6 | Технічний персонал | TECH\_  STAFF | Номер співробітника | Відомості про технічний персонал, кадровий склад цеху, ділянки | Обмежений |
| 7 | Робітники | LABORERS | Номер співробітника | Відомості про робочих та їх посади | Обмежений |
| 8 | Бригади | BRIGADES | ID бригади | Перелік бригад | Загальний |

* 1. **Визначення атрибутів**

Таблиця атрибутів 4. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Найменування | Умовне позначення | Тип | Формат | NULL | Ознака |
| Цехи | | | | | | |
| 1 | Номер\_цеху | ID\_workshop | Атомарний | int | NOT NULL | Primary key |
| 2 | Назва цеху | Workshop\_Name | Атомарний | varchar(50) | NOT NULL | Unique |
| 3 | Номер будівлі | Bulding\_num | Атомарний | smallint | NOT NULL |  |
| Ділянки | | | | | | |
| 1 | Номер\_ділянки | ID\_department | Атомарний | int | NOT NULL | Primary key |
| 2 | Номер\_цеху | ID\_workshop | Атомарний | int | NOT NULL | Foreign key |
| 3 | Назва ділянки | Department\_Name | Атомарний | varchar | NOT NULL |  |
| Вироби | | | | | | |
| 1 | ID\_вироба | ID\_product | Атомарний | int identity(1,1) | NOT NULL | Primary key |
| 2 | Назва | Product\_name | Атомарний | varchar(50) | NOT NULL |  |

Продовження таблиці 4. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Найменування | Умовне позначення | Тип | Формат | NULL | Ознака |
| Категорії літаків | | | | | | |
| 1 | ID\_категорії | ID\_category | Атомарний | int identity(1,1) | NOT NULL | Primary key |
| 2 | Тип | TypeOf | Атомарний | varchar(50) | NOT NULL | check |
| 3 | Число двигунів | AmountOfEngines | Атомарний | tinyint | NULL |  |
| 4 | Матеріал | Material | Багатозначний | varchar(100) | NULL |  |
| 5 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| Категорії дельтапланів | | | | | | |
| 1 | ID\_категорії | ID\_category | Атомарний | Int identity(1,1) | NOT NULL | Primary key |
| 2 | Тип | TypeOf | Атомарний | varchar(50) | NOT NULL | check |
| 3 | Вид дельтаплану | KindHangGlider | Атомарний | varchar(50) | NULL |  |
| 5 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| Категорії гвинтокрилів | | | | | | |
| 1 | ID\_категорії | ID\_category | Атомарний | Int identity(1,1) | NOT NULL | Primary key |
| 2 | Тип | TypeOf | Атомарний | varchar(50) | NOT NULL | check |
| 3 | К-сть взмахів | AmountOfSwings | Атомарний | int | NULL |  |
| 5 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| Бригади | | | | | | |
| 1 | ID\_бригади | ID\_brigade | Атомарний | int identity(1,1) | NOT NULL | Primary key |
| 2 | Номер бригади | BrigadeNum | Атомарний | tinyint | NOT NULL |  |

Продовження таблиці 4. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Найменування | Умовне позначення | Тип | Формат | NULL | Ознака |
| Категорії ракет | | | | | | |
| 1 | ID\_категорії | ID\_category | Атомарний | Int identity(1,1) | NOT NULL | Primary key |
| 2 | Тип | TypeOf | Атомарний | varchar(50) | NOT NULL | check |
| 3 | Потужність заряду | ChargeCapacity | Атомарний | int | NULL |  |
| 4 | Дальність влучення | Length | Атомарний | int | NULL |  |
| 5 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| Категорії планерів | | | | | | |
| 1 | ID\_категорії | ID\_category | Атомарний | int | NOT NULL | Primary key |
| 2 | Тип | TypeOf | Атомарний | varchar(50) | NOT NULL | check |
| 3 | Матеріал крил | MaterialOfWings | Багатозначний | varchar(100) | NULL |  |
| 5 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| Робочі | | | | | | |
| 1 | Номер\_робітника | ID\_laborer | Атомарний | int identity(1,1) | NOT NULL | Primary key |
| 2 | П.І.Б. | Full\_name | Багатозначний | varchar(100) | NOT NULL |  |
| 5 | Дата народження | Date\_birth | Атомарний | date | NOT NULL |  |
| 6 | Спеціалізація | Specialization | Багатозначний | varchar(100) | NOT NULL | Check |
| 7 | Посада | Position | Атомарний | varchar(50) | NOT NULL | Check |
| 8 | Зарплата | Salary | Атомарний | money | NOT NULL |  |
| 9 | Бригада | ID\_brigade | Атомарний | int | NOT NULL | Foreign key |

Продовження таблиці 4. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Найменування | Умовне позначення | Тип | Формат | NULL | Ознака |
| Технічний персонал | | | | | | |
| 1 | Номер\_робітника | ID\_laborer | Атомарний | int identity(1,1) | NOT NULL | Primary key |
| 2 | ПІБ | Full\_name | Багатозначний | varchar(100) | NOT NULL |  |
| 3 | Дата народження | Date\_birth | Атомарний | date | NOT NULL |  |
| 4 | Спеціалізація | Specialization | Багатозначний | varchar(100) | NOT NULL | Check |
| 5 | Посада | Position | Атомарний | varchar(50) | NOT NULL | Check |
| 6 | Зарплата | Salary | Атомарний | money | NOT NULL |  |
| 7 | Цех або ділянка | ID\_department  ID\_workshop | Атомарний | int | NULL | Foreign key |
| Виготовлення виробів | | | | | | |
| 1 | ID\_роботи | ID\_work | Атомарний | int | NOT NULL | Primary key |
| 2 | ID\_вироба | ID\_product | Атомарний | int | NOT NULL | Foreign key |
| 3 | ID\_ділянки | ID\_department | Атомарний | int | NOT NULL | Foreign key |
| 4 | ID\_бригади | ID\_brigade | Атомарний | int | NOT NULL | Foreign key |
| 5 | Дата початку роботи | Start\_date | Атомарний | Date | NOT NULL |  |
| 6 | Дата кінця роботи | End\_date | Атомарний | Date | NULL |  |
| 7 | Стан роботи | Progress | Атомарний | varchar(50) | NULL | Check |

**4.3 Визначення зв’язків**

Таблиця зв'язків 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування | Тип зв'язку | Характер | Обов'язковість |
| 1 | Бригади складають робітники | Ідентифікуючий | Один до багатьох | Бригада повинна містити щонайменше 1 робітника, робітник необов’язково повинен бути в бригаді. |
| 2 | Цехом керує технічний персонал | Неідентифікуючий | Один до необов’язково багатьох | Начальник цеху обирається обов’язково з тех. персоналу, останній необов’язково належить до цеху |
| 3 | Ділянкою керує технічний персонал | Неідентифікуючий | Необов’язково один до багатьох | Начальник ділянки обирається обов’язково з тех. персоналу, останній обов’язково належить до ділянки |
| 4 | Цехи поділяються на ділянки | Неідентифікуючий | Один до багатьох | Цехи обов’язково поділяються на ділянки, ділянки стосуються конкретного цеху |

Продовження таблиці 4. 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування | Тип зв'язку | Характер | Обов'язковість |
| 5 | На Ділянки відбувається виготовлення виробів | Ідентифікуючий | Багато до багатьох | Виготовлення обов’язково відбувається на ділянці, необов’язково на одній |
| 6 | Бригади працюють над виготовленням виробів | Ідентифікуючий | Один до багатьох | Бригада працює лише на одній ділянці, необов’язково над одним виробом |
| 7 | Вироби мають категорію | Неідентифікуючий | Багато до багатьох | Категорія обов’язково відповідає виробу, а вироби можуть існувати без категорії |
| 8 | Вироби в процесі виготовлення | Ідентифікуючий | Багато до необов’язково багатьох | Виготовляють обов’язково вироби і навпаки. |

**4.4 Результати концептуального моделювання**

Основними користувачами є начальники цехів, начальники ділянок та бригадири, оскільки вони є керуючою основою усього підприємства загалом. Бригадири мають права щодо формування бригад з «Робочі», але не мають права редагувати роботу на ділянці та над виробом. Начальники ділянок мають право редагувати «Ділянки» та обирати майстрів у підпорядкування з «Технічний персонал», Встановлюють «Бригади» на конкретну «Ділянки» у «Виготовлення виробів». Начальники цехів мають дозвіл на редагування та додавання «Вироби» та «Категорії», встановлювати «Вироби» у «Виготовлення виробів», редагувати «Технічний персонал» та «Робочі».

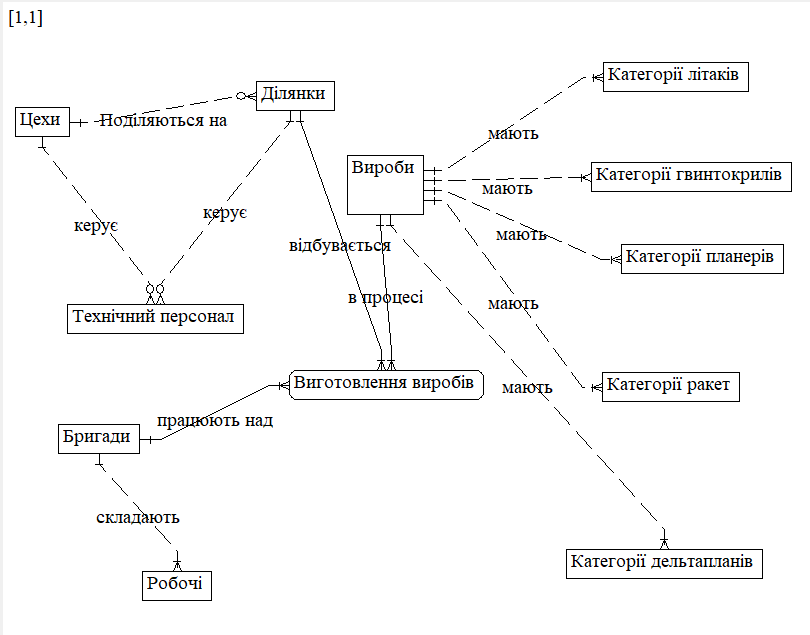


Рисунок 4. 1 ER-діаграма «сутність – зв’язок»

**5. Реляційна модель бази даних**

За результатами Таблиці 4.1, Таблиці 4.2 та Таблиці 4.3 зі зазначенням первинних та зовнішніх ключів можна встановити «Виготовлення виробів» як слабку сутність, яка не може існувати без інших. Інші сутності можна віднести до категорії «сильних».

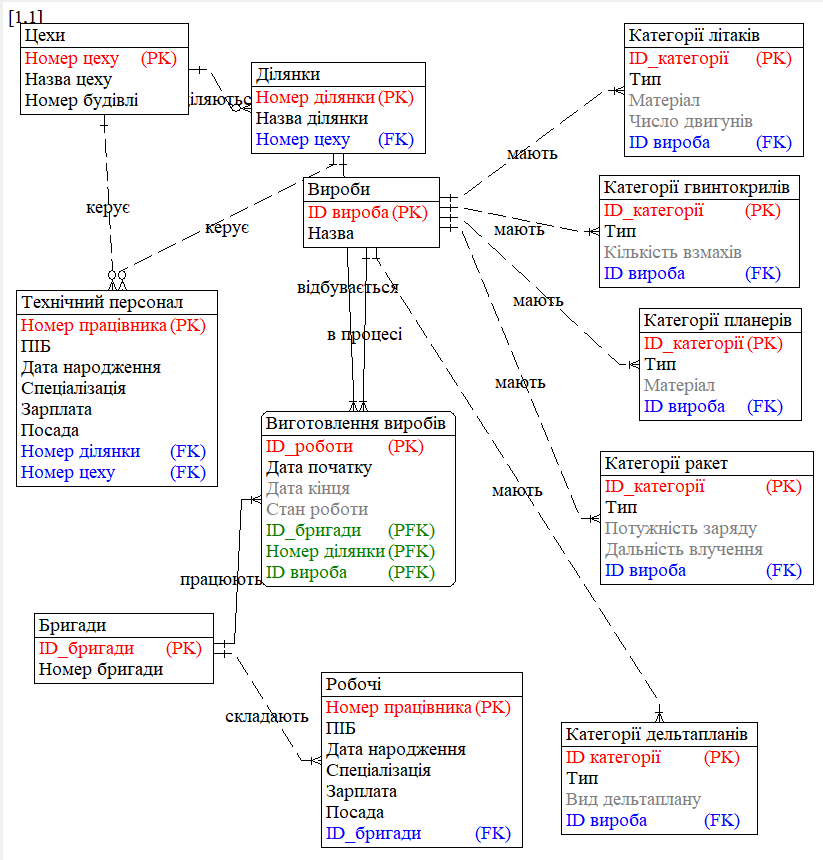


Рисунок 5. 1 Реляційна модель

**6. Фізична модель бази даних**

Після отримання реляційної моделі бази даних час створити її фізичну модель за допомогою MS SQL Server, ссилаючись на таблиці 4.1, 4.2 та 4.3. Одразу присвоюю зовнішні ключі та обмеження на найменування.

**6.1 Створення таблиць**

Таблиці, які відповідають за структуру підприємства:

CREATE TABLE WORKSHOPS (

ID\_WORKSHOP INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

WORKSHOP\_NAME VARCHAR(256) UNIQUE NOT NULL,

BUILDING\_NUM SMALLINT NOT NULL);

CREATE TABLE DEPARTMENTS (

ID\_DEPARTMENT INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

DEPARTMENT\_NAME VARCHAR(256) NOT NULL,

ID\_WORKSHOP INT NOT NULL,

CONSTRAINT FK\_ID\_WORKSHOP FOREIGN KEY (ID\_WORKSHOP) REFERENCES WORKSHOPS (ID\_WORKSHOP));

Таблицям, які відповідають за вироби та категорії додаю обмеження на найменування типів об’єктів, які виробляє підприємство, а на зовнішній ключ ставлю каскадне видалення.

CREATE TABLE PRODUCTS (

ID\_PRODUCT INT IDENTITY(1,1) UNIQUE NOT NULL PRIMARY KEY,

PRODUCT\_NAME VARCHAR(50) NOT NULL);

CREATE TABLE CATEGORIES\_OF\_PLANES (

ID\_CATEGORY INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TYPEOF VARCHAR(50) NOT NULL,

AMOUNT\_OF\_ENGINES TINYINT NULL,

MATERIAL VARCHAR(100) NULL,

ID\_PRODUCT INT NOT NULL,

CONSTRAINT CHECK\_TYPE\_PLANE CHECK(TYPEOF IN ('Civil', 'Transport', 'Military')),

CONSTRAINT FK\_ID\_PROD1 FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT) ON DELETE CASCADE);

CREATE TABLE CATEGORIES\_OF\_PLANNERS (

ID\_CATEGORY INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TYPEOF VARCHAR(50) NOT NULL,

MATERIAL\_OF\_WINGS VARCHAR(56) NULL,

ID\_PRODUCT INT NOT NULL,

CONSTRAINT CHECK\_TYPE\_PLANNERS CHECK(TYPEOF IN ('For-two', 'Mechanized', 'Military')),

CONSTRAINT FK\_ID\_PROD2 FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT) ON DELETE CASCADE);

CREATE TABLE CATEGORIES\_OF\_HELICOPTERS (

ID\_CATEGORY INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TYPEOF VARCHAR(50) NOT NULL,

AMOUNT\_OF\_SWINGS INT NULL,

ID\_PRODUCT INT NOT NULL,

CONSTRAINT CHECK\_TYPE\_HELICOPTERS CHECK(TYPEOF IN ('Military', 'Civil', 'Single-screw', 'Double-srew')),

CONSTRAINT FK\_ID\_PROD3 FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT) ON DELETE CASCADE);

CREATE TABLE CATEGORIES\_OF\_ROCKETS (

ID\_CATEGORY INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TYPEOF VARCHAR(50) NOT NULL,

CHARGE\_CAPACITY INT NULL,

LENGTHOF INT NULL,

ID\_PRODUCT INT NOT NULL,

CONSTRAINT CHECK\_TYPE\_ROCKETS CHECK(TYPEOF IN ('Artillery', 'Aviation', 'Military', 'Naval')),

CONSTRAINT FK\_ID\_PROD4 FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT) ON DELETE CASCADE);

CREATE TABLE CATEGORIES\_OF\_HANG\_GLIDERS (

ID\_CATEGORY INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TYPEOF VARCHAR(50) NOT NULL,

KIND\_OF\_HANGGLIDER VARCHAR(50) NULL,

ID\_PRODUCT INT NOT NULL,

CONSTRAINT CHECK\_TYPE\_GLIDERS CHECK(TYPEOF IN ('Sport', 'Special purpose')),

CONSTRAINT FK\_ID\_PROD5 FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT) ON DELETE CASCADE);

Таблицям, що стосуються кадрового складу підприємства створюю обмеження на атрибути «SPECIALIZATION» та «POSITION», оскільки підприємство чітко моделює: робітники якого типу їм потрібні та які посади існують. Таким чином у майбутньому можна легко змоделювати хто є хто у кадровому складі підприємства.

CREATE TABLE BRIGADES (

ID\_BRIGADE INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

BRIGADE\_NUM TINYINT NOT NULL);

CREATE TABLE LABORERS (

ID\_LABORER INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

FULL\_NAME VARCHAR(256) NOT NULL,

DATE\_BIRTH DATE NOT NULL,

SPECIALIZATION VARCHAR(100) NOT NULL CHECK(SPECIALIZATION IN ('Picker', 'Turner', 'Fitter', 'Welder')),

SALARY MONEY NOT NULL,

POSITION VARCHAR(50) NOT NULL CHECK(POSITION IN('Brigader', 'Laborer')),

ID\_BRIGADE INT NULL,

CONSTRAINT FK\_ID\_BRIGADE1 FOREIGN KEY (ID\_BRIGADE) REFERENCES BRIGADES (ID\_BRIGADE));

CREATE TABLE TECH\_STAFF (

ID\_LABORER INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

FULL\_NAME VARCHAR(256) NOT NULL,

DATE\_BIRTH DATE NOT NULL,

SPECIALIZATION VARCHAR(100) NOT NULL CHECK(SPECIALIZATION IN ('Engineer', 'Technologist', 'Technician')),

SALARY MONEY NOT NULL,

POSITION VARCHAR(50) NOT NULL CHECK(POSITION IN ('Head of workshop', 'Head of department', 'Master')),

ID\_WORKSHOP INT NULL,

ID\_DEPARTMENT INT NULL,

CONSTRAINT FK\_ID\_WORKSHOP2 FOREIGN KEY (ID\_WORKSHOP) REFERENCES WORKSHOPS (ID\_WORKSHOP),

CONSTRAINT FK\_ID\_DEPARTMENT2 FOREIGN KEY (ID\_DEPARTMENT) REFERENCES DEPARTMENTS (ID\_DEPARTMENT)

);

Слабка сутність «PROGRESS» складається із зовнішніх ключів та інформації щодо роботи. Я не ставлю обмеження на діапазон значень атрибута «Progress», оскільки в будь який момент можуть бути внесені відмінні значення від «Планується», «Виконано» та «В процесі».

CREATE TABLE PROGRESS (

ID\_WORK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

ID\_PRODUCT INT NOT NULL,

ID\_DEPARTMENT INT NOT NULL,

ID\_BRIGADE INT NOT NULL,

START\_\_DATE DATE NOT NULL,

END\_\_DATE DATE NULL,

PROGRESS VARCHAR(50) NULL,

CONSTRAINT FK\_ID\_PRODUCT FOREIGN KEY (ID\_PRODUCT) REFERENCES PRODUCTS (ID\_PRODUCT),

CONSTRAINT FK\_ID\_WORKSHOP3 FOREIGN KEY (ID\_DEPARTMENT) REFERENCES DEPARTMENTS (ID\_DEPARTMENT),

CONSTRAINT FK\_ID\_BRIGADE2 FOREIGN KEY (ID\_BRIGADE) REFERENCES BRIGADES (ID\_BRIGADE));

**6.2 Забезпечення цілісності бази даних**

Задля запобігання цілісності створеної бази даних додаю триггери на деякі значення.

По-перше, дата кінця роботи на ділянці не може бути більшою за дату початку роботи: створюю триггер.

CREATE TRIGGER WORK\_DATE\_CHANGE

ON PROGRESS

FOR UPDATE, INSERT

AS

IF EXISTS

(SELECT 'TRUE'

FROM PROGRESS

WHERE START\_\_DATE>END\_\_DATE)

BEGIN

RAISERROR('DATE OF WORK`S END CAN`T BE BIGGER THAN THE START DATE', 16, 1);

ROLLBACK TRANSACTION;

END

По-друге, для запобігання збереження інформації про виготовлення виробів, створила триггер, що забороняє видалення записів про виготовлені вироби.

CREATE TRIGGER WORK\_DATA\_DELETE

ON PROGRESS

FOR DELETE

AS

IF EXISTS(

SELECT 'TRUE'

FROM PROGRESS

WHERE PROGRESS='Completed')

BEGIN

RAISERROR('INFORMATION CAN`T BE DELETED', 16, 1);

ROLLBACK TRANSACTION;

END;

Робітники на одних і тих самих позиціях отримують однакову зарплатню, тож створимо триггер, який незважаючи на дані, які ми внесемо до атрибута «SALARY» буде оновлювати зарплатню згідно з посадою робітника.

CREATE TRIGGER LABORS\_SALARY

ON LABORERS

AFTER UPDATE, INSERT

AS

BEGIN

UPDATE LABORERS

SET SALARY=10000 WHERE POSITION='Laborer'

UPDATE LABORERS

SET SALARY=15000 WHERE POSITION='Brigader'

END

Те саме робимо для таблиці із даними про Технічний персонал, зважаючи на їх посаду: Начальник цеху, начальник ділянки або майстер.

CREATE TRIGGER TECHSTAFF\_SALARY

ON TECH\_STAFF

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE TECH\_STAFF

SET SALARY=15000 WHERE POSITION='Master'

UPDATE TECH\_STAFF

SET SALARY=20000 WHERE POSITION='Head of departmnet'

UPDATE TECH\_STAFF

SET SALARY=25000 WHERE POSITION='Head of workshop'

END

Створюючи таблицю з даними про технічний персонал, треба зауважити, що лише майстри та начальники ділянок стосуються ділянок, а начальники цехів – цехів.

CREATE TRIGGER POSITION\_TECH\_STAFF

ON TECH\_STAFF

FOR INSERT, UPDATE

AS

IF EXISTS(

SELECT 'TRUE'

FROM INSERTED

WHERE (POSITION='Head of department' AND ID\_WORKSHOP IS NOT NULL)

OR (POSITION='Master' AND ID\_WORKSHOP IS NOT NULL)

OR (POSITION='Head of workshop' AND ID\_DEPARTMENT IS NOT NULL))

BEGIN

RAISERROR('Information is incorrect. Check technician staff positions!', 16, 1)

ROLLBACK TRANSACTION;

END

Якщо замислитися, що буде з робітниками, якщо бригаду ліквідують? Їх не звільнять, вони будуть чекати формування нової бригади. Саме тому після видалення треба автоматично присвоїти їхньому атрибуту «ID\_BRIGADE» значення «NULL».

CREATE TRIGGER LABORERSINBRIGADES

ON BRIGADES

INSTEAD OF DELETE

AS

BEGIN

DECLARE @ID\_BRIGADE INT

SELECT @ID\_BRIGADE=D.ID\_BRIGADE

FROM DELETED D

IF EXISTS

(SELECT 'TRUE'

FROM DELETED D

INNER JOIN LABORERS L

ON D.ID\_BRIGADE=L.ID\_BRIGADE)

UPDATE LABORERS

SET ID\_BRIGADE=NULL WHERE ID\_BRIGADE=@ID\_BRIGADE

DELETE FROM BRIGADES WHERE ID\_BRIGADE=@ID\_BRIGADE

END

**6.3 Приклад заповненої бази даних**

Заповнюючи таблиці створеної бази даних, ураховую всі встановлені обмеження, використовуючи скрипти .sql.

INSERT INTO WORKSHOPS(WORKSHOP\_NAME,BUILDING\_NUM) VALUES ('PROCESSING', 216),('FOLDING', 155);

INSERT INTO DEPARTMENTS(DEPARTMENT\_NAME, ID\_WORKSHOP) VALUES ('Design', 1), ('Production of materials', 1), ('Assembly', 1), ('Acceptance of delivery', 2), ('Assembly', 2), ('Reconstruction', 2), ('Test run', 2);

INSERT INTO PRODUCTS(PRODUCT\_NAME) VALUES ('F-22 Raptor boeing'), ('An-22'), ('Ant-20'), ('Taurus'), ('Airspeed Horsa'), ('Slyline SL-222'), ('AK1-3'), ('Neptun P-360'), ('Alta'), ('P-27'), ('Aeros'), ('ATOS VQrace');

INSERT INTO CATEGORIES\_OF\_PLANES(TYPEOF, AMOUNT\_OF\_ENGINES, MATERIAL, ID\_PRODUCT) VALUES ('Military', 2, 'Carbon fiber', 1), ('Transport', 4, 'Aluminium', 2), ('Civil', 8, 'Corrugation', 3);

INSERT INTO CATEGORIES\_OF\_PLANNERS(TYPEOF, MATERIAL\_OF\_WINGS, ID\_PRODUCT) VALUES ('For-two', 'Zinc', 4), ('Military', 'Metal', 5);

INSERT INTO CATEGORIES\_OF\_HELICOPTERS(TYPEOF, AMOUNT\_OF\_SWINGS, ID\_PRODUCT) VALUES ('Civil', 2, 6), ('Double-srew', 2, 7);

INSERT INTO CATEGORIES\_OF\_ROCKETS(TYPEOF, CHARGE\_CAPACITY, LENGTHOF, ID\_PRODUCT) VALUES ('Naval', 0.8, 280, 8), ('Artillery', 0.8, 12, 9), ('Aviation', 4.5, 110, 10);

INSERT INTO CATEGORIES\_OF\_HANG\_GLIDERS(TYPEOF, KIND\_OF\_HANGGLIDER, ID\_PRODUCT) VALUES ('Special purpose', 'Super light', 11), ('Sport', 'Maneuvering', 12);

INSERT INTO BRIGADES (BRIGADE\_NUM) VALUES (204), (33), (67), (105), (18);

INSERT INTO LABORERS (FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, SALARY, ID\_BRIGADE) VALUES

('Petrenko Petr Petrovych', '2000-11-05', 'Picker', 'Laborer', 10000, 1),

('Williams Alex Sergeev', '1995-02-11', 'Turner', 'Brigader', 15000, 1),

('Evans Susane Aleksandrovna', '1998-06-12', 'Fitter', 'Laborer', 10000, 2),

('Gibson John Artemovich', '1977-10-25', 'Welder', 'Brigader', 15000, 2),

('Campbell Miranda Engenevna', '1999-03-12', 'Turner', 'Laborer', 10000, 2),

('Stetsenko Kirill Pavlovich', '1995-10-20', 'Picker', 'Brigader', 15000, 3),

('Karp Oleh Vyacheslavovich', '1989-05-10', 'Turner', 'Laborer', 10000, 3),

('Zhema Vlad Vadomivich', '1991-09-03', 'Welder', 'Brigader', 15000, 4),

('Adams Pavlo Dmotrovich', '1990-11-15', 'Fitter', 'Laborer', 10000, 4),

('Alushta Gerald Vladimirovich', '1999-02-09', 'Picker', 'Brigader', 15000, 5),

('Matvienko Leonid Stanislavovich', '1978-12-01', 'Turner', 'Laborer', 10000, 5);

INSERT INTO TECH\_STAFF (FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, SALARY, ID\_DEPARTMENT) VALUES

('Marlov Artem Semenovych', '1978-12-12', 'Engineer', 'Master', 15000, 1),

('Mizuka Sam Valeriyovych', '1988-10-02', 'Engineer', 'Head of department', 20000, 1),

('Kozhumaka Ivan Fedorovych', '1999-08-01', 'Technologist', 'Master', 15000, 2),

('Semenko Anatolyi Petrivych', '1995-10-20', 'Technologist', 'Head of department', 20000, 2),

('Garpun Matiya Sergiyivna', '1998-11-27', 'Technician', 'Master', 15000, 3),

('Sergeev Volodymyr Maksymovych', '2000-01-07', 'Technician', 'Head of department', 20000, 3),

('Tan Maksym Bogdanovych', '1996-10-15', 'Technologist', 'Head of department', 20000, 4),

('Shestak Maksym Vadimovych', '1999-10-15', 'Engineer', 'Master', 15000, 4),

('Bigday Lubov Myronivna', '1997-05-27', 'Engineer', 'Head of department', 20000, 5),

('Sonnik Dmytro Leonidovych', '1977-12-21', 'Technologist', 'Master', 15000, 5),

('Polishschuk Olexandra Sergiyivna', '1994-10-30', 'Technician', 'Head of department', 20000, 6),

('Marzhuk Denyd Yakovlevych', '1991-12-25', 'Engineer', 'Master', 15000, 6),

('Lyka Semen Borysovych', '1999-06-22', 'Technician', 'Master', 15000, 7),

('Stupak Oleg Mykolaevych', '1994-02-20', 'Technologist', 'Head of department', 20000, 7)

INSERT INTO TECH\_STAFF(FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, SALARY, ID\_WORKSHOP) VALUES

('Vinnyk Oleg Hrygorovych', '1989-02-27', 'Technologist', 'Head of workshop', 25000, 1),

('Kozh Stepan Dmytrovych', '1985-10-15', 'Engineer', 'Head of workshop', 25000, 2)

INSERT INTO PROGRESS(ID\_PRODUCT, ID\_DEPARTMENT, ID\_BRIGADE, START\_\_DATE, END\_\_DATE, PROGRESS) VALUES

(1, 1, 1, '2022-02-10', '2022-10-10', 'Completed'),

(1, 2, 2, '2022-10-11', NULL, 'In progress'),

(2, 3, 3, '2020-11-09', NULL, 'In progress'),

(3, 1, 4, '2023-01-10', NULL, 'Planning'),

(4, 2, 4, '2022-11-10', NULL, 'In progress'),

(5, 3, 2, '2021-10-9', '2022-01-05', 'Completed'),

(6, 4, 5, '2021-05-10', '2021-05-15', 'Completed'),

(6, 5, 5, '2021-06-01', '2021-07-30', 'Completed'),

(7, 6, 5, '2022-12-25', NULL, 'Planning'),

(8, 5, 4, '2021-12-10', NULL, 'Postponed'),

(9, 7, 1, '2020-10-10', '2020-10-11', 'Completed'),

(10, 6, 2, '2019-11-12', '2019-11-30', 'Completed'),

(11, 3, 4, '2023-10-11', NULL, 'Planning'),

(12, 2, 3, '2023-01-10', NULL, 'Planning')

**7. Реалізація завдань**

Задля того, щоб якісніше та простіше реалізувати роботи деяких запитів в подальшому будемо створювати процедури, які будуть повертати інформацію за допомогою зазначених параметрів.

1. Отримати перелік видів виробів окремої категорії і в цілому, що збираються зазначеним цехом, підприємством.

Створюємо процедуру, яка буде виводити вироби та їх характеристику при зазначенні назви цеху(тобто підприємства). Для цього атрибути з таблиць, що відповідають за категорії виробів, поєднуємо в один атрибут «Характеристики» за допомогою функції CONVERT. Усі таблиці знаходяться в підзапиті, для того, щоб для всіх приймалися однакові правила UNION.

CREATE PROC WORKSHOP\_PRODUCTS

@WORKSHOP\_NAME VARCHAR(50)

AS

SELECT DISTINCT CAT.TYPE, PRODUCT\_NAME AS 'PRODUCT', TYPEOF AS 'CATEGORY', CHARACTERISTIC, WORKSHOP\_NAME AS 'WORKSHOP' FROM

(SELECT WORKSHOP\_NAME, 'Plane' AS TYPE, PRODUCT\_NAME, TYPEOF, CONVERT(VARCHAR(50), AMOUNT\_OF\_ENGINES) +' engines, '+ CONVERT(VARCHAR(50), MATERIAL) + ' material' AS CHARACTERISTIC

FROM CATEGORIES\_OF\_PLANES PL

INNER JOIN PRODUCTS PRO

ON PL.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN PROGRESS P

ON P.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

UNION

SELECT WORKSHOP\_NAME, 'Helicopter' AS TYPE, PRODUCT\_NAME, TYPEOF, CONVERT(VARCHAR(50), AMOUNT\_OF\_SWINGS) +' swings' as CHARACTERISTIC

FROM CATEGORIES\_OF\_HELICOPTERS PL INNER JOIN PRODUCTS PRO ON PL.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN PROGRESS P

ON P.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

UNION

SELECT WORKSHOP\_NAME, 'Planner' AS TYPE, PRODUCT\_NAME, TYPEOF, CONVERT(VARCHAR(50), MATERIAL\_OF\_WINGS) + ' material pf wings' as CHARACTERISTIC

FROM CATEGORIES\_OF\_PLANNERS PL INNER JOIN PRODUCTS PRO ON PL.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN PROGRESS P

ON P.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

UNION

SELECT WORKSHOP\_NAME, 'Rocket' AS TYPE, PRODUCT\_NAME, TYPEOF, CONVERT(VARCHAR(50), CHARGE\_CAPACITY) + ' charge capacity, ' +

CONVERT(VARCHAR(50), LENGTHOF) + ' length' AS CHARACTERISTIC

FROM CATEGORIES\_OF\_ROCKETS PL INNER JOIN PRODUCTS PRO ON PL.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN PROGRESS P

ON P.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

UNION

SELECT WORKSHOP\_NAME, 'Hang glider' AS TYPE, PRODUCT\_NAME, TYPEOF, CONVERT(VARCHAR(50), KIND\_OF\_HANGGLIDER) AS CHARACTERISTIC

FROM CATEGORIES\_OF\_HANG\_GLIDERS PL

INNER JOIN PRODUCTS PRO

ON PL.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN PROGRESS P

ON P.ID\_PRODUCT=PRO.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP) CAT

WHERE CAT.WORKSHOP\_NAME=@WORKSHOP\_NAME

**Приклад реалізації процедури:**

EXEC WORKSHOP\_PRODUCTS 'PROCESSING'

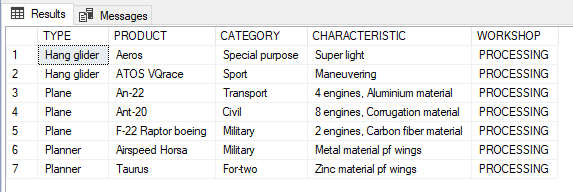


Рисунок 7. 1

2. Отримати число і перелік виробів окремої категорії і в цілому, зібраних зазначеним цехом, ділянкою, підприємством в цілому за певний відрізок часу.

Створюємо процедуру, яка при внесені початкової та кінцевої дати буде виводи виготовлені вироби: для цього вносимо змінну початкової дати, яка повинна бути меншою, або дорівнювати атрибути START\_\_DATE та змінну кінцевої дати, яка повинна бути більшою, або дорівнювати END\_\_DATE. Таким чином ми створюємо відрізок с початковою та кінцевою точками.

CREATE PROC PRODUCTS\_TIME\_PRODUCTION

@START\_\_DATE DATE,

@END\_\_DATE DATE

AS

SELECT DISTINCT PRODUCT\_NAME AS 'PRODUCT', DEPARTMENT\_NAME AS 'DEPARTMENT', WORKSHOP\_NAME AS 'WORKSHOP', START\_\_DATE AS 'START OF WORK', END\_\_DATE AS 'END OF WORK', PROGRESS

FROM

PROGRESS PRO

INNER JOIN PRODUCTS DU

ON PRO.ID\_PRODUCT=DU.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON PRO.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

WHERE PRO.START\_\_DATE>=@START\_\_DATE AND PRO.END\_\_DATE<=@END\_\_DATE

**Приклад реалізації процедури:**

EXEC PRODUCTS\_TIME\_PRODUCTION '2020-10-10', '2022-12-28'

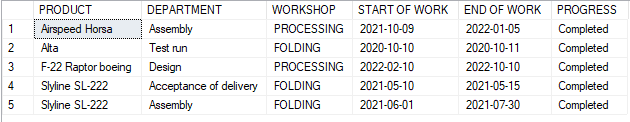


Рисунок 7. 2

3. Отримати дані про кадровий склад цеху, підприємства в цілому і по зазначеним категоріям інженерно-технічного персоналу і робітників.

Задля створення запиту використовуємо UNION для поєднання робітників, що стосуються різних структур підприємства.

SELECT FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, CONVERT(VARCHAR(50), BRIGADE\_NUM) + ' BRIGADE' AS 'OFFICE'

FROM LABORERS L

INNER JOIN BRIGADES B

ON L.ID\_BRIGADE=B.ID\_BRIGADE

UNION

SELECT FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, CONVERT(VARCHAR(50), DEPARTMENT\_NAME) + ' DEPARTMENT' AS OFFICE

FROM TECH\_STAFF TE INNER JOIN

DEPARTMENTS D

ON TE.ID\_DEPARTMENT=D.ID\_DEPARTMENT

UNION

SELECT FULL\_NAME, DATE\_BIRTH, SPECIALIZATION, POSITION, CONVERT(VARCHAR(50), WORKSHOP\_NAME) + ' WORKSHOP' AS OFFICE

FROM TECH\_STAFF TE

INNER JOIN WORKSHOPS W

ON TE.ID\_WORKSHOP=W.ID\_WORKSHOP

**Приклад реалізації запиту:**

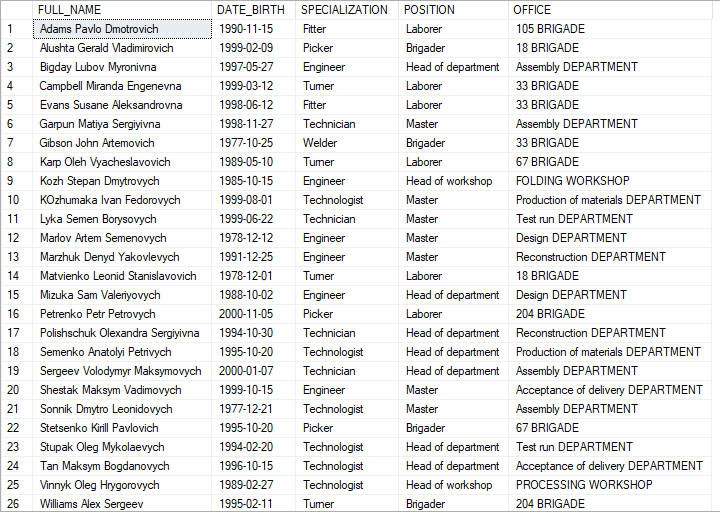
****

Рисунок 7. 3

4. Отримати число і перелік ділянок зазначеного цеху, підприємства в цілому і їх начальників.

Створюємо процедуру, яка буде виводи ділянки та їх начальників при зазначені цеху. В умові вказуємо, що посада робітника повинна бути «Начальник ділянки».

CREATE PROC FACTORY\_STRUCTURE

@WORKSHOP\_NAME VARCHAR(50)

AS

SELECT DEPARTMENT\_NAME AS 'DEPARTMENT', FULL\_NAME AS 'HEAD OF DEPARTMENT'

FROM DEPARTMENTS D

INNER JOIN TECH\_STAFF TE

ON D.ID\_DEPARTMENT=TE.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

WHERE POSITION='Head of department' AND WORKSHOP\_NAME=@WORKSHOP\_NAME

**Приклад реалізації процедури:**

EXEC FACTORY\_STRUCTURE 'PROCESSING'

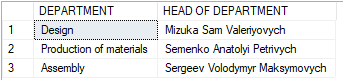


Рисунок 7. 4

5. Отримати перелік робіт, які проходить вказаний виріб.

Створюємо процедуру, яка буде повертати дані про процеси в яких задіяний, або був задіяним виріб.

CREATE PROC PRODUCT\_PROGRESS

@PRODUCT\_NAME VARCHAR(50)

AS

SELECT PRODUCT\_NAME, DEPARTMENT\_NAME, START\_\_DATE, END\_\_DATE, PROGRESS

FROM PROGRESS PRO

INNER JOIN PRODUCTS P

ON PRO.ID\_PRODUCT=P.ID\_PRODUCT

INNER JOIN DEPARTMENTS D

ON PRO.ID\_DEPARTMENT=D.ID\_DEPARTMENT

WHERE PRODUCT\_NAME=@PRODUCT\_NAME

**Приклад реалізації процедури:**

EXEC PRODUCT\_PROGRESS 'Slyline SL-222'

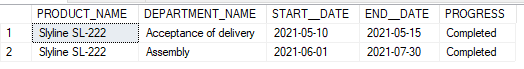


Рисунок 7. 5

6. Отримати склад бригад вказаної ділянки, цеху.

Створюємо процедуру, яка буде виводи склад бригад, які були задіяні у зазначеній ділянці.

CREATE PROC BRIGADE\_STRUCTURE

@DEPARTMENT\_NAME VARCHAR(50)

AS

SELECT WORKSHOP\_NAME, DEPARTMENT\_NAME, BRIGADE\_NUM, FULL\_NAME, POSITION

FROM DEPARTMENTS D

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

INNER JOIN PROGRESS P

ON P.ID\_DEPARTMENT=D.ID\_DEPARTMENT

INNER JOIN BRIGADES B

ON P.ID\_BRIGADE=B.ID\_BRIGADE

INNER JOIN LABORERS L

ON B.ID\_BRIGADE=L.ID\_BRIGADE

WHERE DEPARTMENT\_NAME=@DEPARTMENT\_NAME

ORDER BY BRIGADE\_NUM

**Приклад реалізації процедури:**

EXEC BRIGADE\_STRUCTURE 'Design'

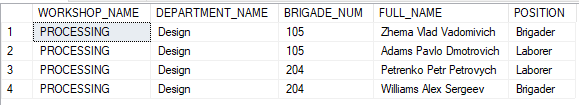


Рисунок 7. 6

7. Отримати список майстрів вказаної ділянки, цеху.

Створюємо процедуру, яка буде виводити лише майстрів зазначеної ділянки.

CREATE PROC MASTER\_LIST

@DEPARTMEN\_NAME VARCHAR(50)

AS

SELECT FULL\_NAME, POSITION, DEPARTMENT\_NAME, WORKSHOP\_NAME

FROM DEPARTMENTS D

INNER JOIN TECH\_STAFF TE

ON D.ID\_DEPARTMENT=TE.ID\_DEPARTMENT

INNER JOIN WORKSHOPS W

ON D.ID\_WORKSHOP=W.ID\_WORKSHOP

WHERE DEPARTMENT\_NAME=@DEPARTMEN\_NAME AND POSITION='Master'

**Приклад реалізації процедури:**

EXEC MASTER\_LIST 'Design'



Рисунок 7. 7

8. Отримати перелік виробів окремої категорії і в цілому, що збираються зараз зазначеною ділянкою, цехом, підприємством.

Створюємо процедуру, яка буде повертати вироби, які зараз в процесі виготовлення.

CREATE PROC DEPARTMENT\_OF\_PRODUCTS\_IN\_PROGRESS

@DEPARTMENT\_NAME VARCHAR(50)

AS

SELECT DEPARTMENT\_NAME, PRODUCT\_NAME, START\_\_DATE

FROM DEPARTMENTS D

INNER JOIN PROGRESS PRO

ON D.ID\_DEPARTMENT=PRO.ID\_DEPARTMENT

INNER JOIN PRODUCTS P

ON PRO.ID\_PRODUCT=P.ID\_PRODUCT

WHERE PROGRESS='In progress' AND DEPARTMENT\_NAME=@DEPARTMENT\_NAME

**Приклад реалізації процедури:**

EXEC DEPARTMENT\_OF\_PRODUCTS\_IN\_PROGRESS 'Production of materials'

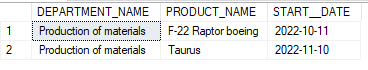


Рисунок 7. 8

9. Отримати склад бригад, що беруть участь в складанні зазначеного виробу.

Створюємо процедуру, яка буде повертати склад бригад вироба, який зараз складається

CREATE PROC PRODUCT\_BRIGADES

@PRODUCT\_NAME VARCHAR(50)

AS

SELECT BRIGADE\_NUM, FULL\_NAME, POSITION

FROM PROGRESS PRO

INNER JOIN PRODUCTS P

ON PRO.ID\_PRODUCT=P.ID\_PRODUCT

INNER JOIN BRIGADES B

ON PRO.ID\_BRIGADE=B.ID\_BRIGADE

INNER JOIN LABORERS L

ON B.ID\_BRIGADE=L.ID\_BRIGADE

WHERE PROGRESS='In progress' AND PRODUCT\_NAME=@PRODUCT\_NAME

**Приклад реалізації процедури:**

EXEC PRODUCT\_BRIGADES 'An-22'



Рисунок 7. 9

**8. Висновки по роботі**

ТУТ БУВ ВИСНОВОК ЯКА Я РОЗУМНА

**Список використаних джерел**

1. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. К.:КНУБА, 2005. - 204 с.
2. Джепаров Д. DFD діаграми – навіщо вони потрібні та які бувають, 2022. URL: <https://habr.com/ru/post/668684/>
3. Кисилев Д. Ю., Кисилев Ю. В., Макар’єв В. Д. Структурний аналіз потоків даних: Методичне видання. Самара: СГАУ, 2014. - 12 с.
4. Створення збереженої процедури: стаття. 2022. URL: <https://learn.microsoft.com/ru-ru/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-ver16>
5. Проблеми розвитку баз даних: стаття. 2022. URL: https://learn.microsoft.com/ru-ru/sql/ssms/visual-db-tools/issues-of-database-evolution-visual-database-tools?view=sql-server-ver16
6. Коломицев М. В. Бази даних. Концептуальне проектування баз даних: лекція. Київ, 2022. URL: <https://do.ipo.kpi.ua/pluginfile.php/457885/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F%203.pdf>
7. Посібник по MS SQL Server 2022. Вкладені запити: стаття. 2022 URL: <https://metanit.com/sql/sqlserver/>