

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
Харківський національний університет радіоелектроніки

Кафедра системотехніки

ЗВІТ

з виконання завдань практичного заняття № 1  
дисципліни «Проектування високонавантажених систем зберігання даних»

Виконав

студент групи ІТКНу-19-2

Марковець Назар Сергійович

Перевірив:

доцент кафедри СТ

Коваленко А.І.

Харків, 2020

## **1 Мета практичного заняття**

Набуття практичних навичок з розробки баз даних на платформі MySQL з таблицями типу MyIsam і InnoDB, з урахуванням особливостей роботи високонавантаженої інформаційної системи зберігання даних.

## **2 Постановки задачі згідно з індивідуальним завданням**

Тема індивідуального завдання: ІС "Надання послуг типографії"

**Завдання 1.1.** Провести аналіз предметної області високонавантаженої інформаційної системи, що складається із серверної (бази даних) і клієнтської (інтерфейсу доступу до бази даних) частин. Для цього необхідно: – дати характеристику діяльності підприємства (організації), що визначається предметною областю; – описати основні бізнес-процеси, які можна автоматизувати з використанням програмних засобів високонавантаженої інформаційної системи; – визначити основні бізнес-функції високонавантаженої інформаційної системи, що відповідають завданням предметної області; – визначити бізнес-функції інтерфейсу клієнтської частини високонавантаженої інформаційної системи, у припущенні, що функціональна частина реалізується на стороні сервера баз даних MySQL; – визначити бізнес-функції інтерфейсу клієнтської частини високонавантаженої інформаційної системи для повнотекстового пошуку з різними параметрами, обумовленими предметною областю.

**Стисла характеристика діяльності підприємства, згідно з індивідуальним завданням.**

Типографія – це спеціалізоване підприємство яке виготовляє поліграфічну продукцію. Самі типографії можуть бути універсальними (такими, на яких проводиться повністю весь процес вироблення поліграфії) та вузькоспеціалізованими (які пропонують лише окремі типи послуг).

Для задоволення клієнтських потреб, важливо ретельно підходити до роботи. Тому вироблення поліграфічної продукції сьогодні прийнято ділити на такі три основні етапи:

- Етап підготовки до друку, на якому споживач узгоджує особливості замовлення з персоналом. Відбувається обговорення тиражу примірників, терміну виконання робіт та вартості послуг інших особливостей (кольори поліграфії: чорно-білий чи повноцінна палітра; матеріал: залежить від категорії поліграфії – папір, картон, текстиль, кераміка, деревина, метал; наповнення та дизайн продукції: де і що буде розташовуватись, яким буде текст, зображення, шрифт).

- Етап створення та друку поліграфічної продукції. Після того, як замовник затвердить макет майбутніх виробів, спеціалісти приступають безпосередньо до їхнього виготовлення. Уся робота відбувається на новітніх потужних принтерах. Виділяють кілька видів друку: офсетний (бюджетний, який дозволяє працювати з невеликою кількістю екземплярів) та цифровий (який найчастіше використовуються для великих тиражів).

- Етап післядруку. Це додаткова опція, яка дозволяє прикрасити поліграфію. На такому етапі відбувається тиснення, висічка, ламінування та інші види робіт.

Сучасні типографії мають налагоджений процес роботи, який дозволяє отримати готовий результат упродовж кількох днів або навіть годин. Варто лиш завчасно продумати дизайн поліграфічної продукції, обрати конкретну категорію та узгодити усі деталі замовлення.

### **Перелік нормативних документів (законних і підзаконних актів), що регламентують діяльність підприємства:**

Діяльність будь-якого друкарського підприємства регламентована законодавчими актами, що прийняті чинним законодавством країни.

- Наказ Про затвердження Державних санітарних норм та правил «Підприємства та організації поліграфічної промисловості»;

- Вимоги Державних санітарних правил та норм;

- Закон України «Про відходи»;

- Постанова Кабінету Міністрів України №596 «Про програму розвитку виробництва газетного та друкарського видів паперу в Україні для задоволення потреб друкованих засобів масової інформації»

- Закон України «Про підприємства в Україні»
- Внутрішній регламент робіт та розпорядку підприємства.

**Перелік програмного забезпечення, необхідного для реалізації трирівневої (триланкової) архітектури «клієнт–сервер» високонавантаженої інформаційної системи:** Visual Studio, браузер, IIS (Internet Information Services), MySQL Shell.

**Перелік можливих типів користувачів високонавантаженої інформаційної системи, їх статусів.**

Таблиця 1.1 – Перелік статусів можливих типів користувачів

№	Користувач	Статус	Доступ до даних (Найменування таблиць)
1	Незареєстрований користувач	Немає	Має доступ до читання таблиці Services
2	Зареєстрований користувач	User	Має доступ до читання та запису таблиць Fonts, Paper, Macket to Print, Order (згідно свого замовлення). Має доступ до читання таблиць Services, User ( тільки власні дані)
3	Адміністратор системи	Admin	Має доступ до запису й читання всіх таблиць БД. Заборонене самовільне редагування таблиці User

**Перелік бізнес-процесів, що підлягають автоматизації:**

- Замовлення необхідного типу послуги з урахуванням всіх доступних особливостей, що може надати типографія
- Реєстрація нових клієнтів

- Занесення додаткових даних ( якщо користувач завантажує свій шрифт або макет)

### **Перелік бізнес-функцій для кожного типу (статусу) користувача:**

Бізнес-функції системи для незареєстрованих користувачів: реєстрація на сайті, перегляд усіх послуг.

Бізнес-функції системи для зареєстрованих користувачів: вхід до системи зі статусом «User»; перегляд усіх доступних послуг, шрифтів, видів паперу, інформацію про власні дані, оформлені замовлення; фільтрація послуг, шрифтів за особливостями; замовлення послуг друку з каталогу для наданого користувачем макету; створення та замовлення особливого замовлення для макету обравши необхідні з доступних послуг; перегляд власних замовлень та їх статусів; оплата замовлення; відмова від замовлення .

Бізнес-функції системи для адміністраторів: вхід в систему з визначенням статусу «Admin»; перегляд користувачів, каталогу послуг, шрифтів, макетів, замовлень; редагування даних каталогу, паперу, шрифтів; зміна статусу замовлення.

**Таблиця з переліком елементів інтерфейсу й пов'язаних з ним бізнес функцій високонавантаженої інформаційної системи, що виконуються на стороні сервера MySQL (включаючи бізнес-функції повнотекстового пошуку).**

Таблиця 1.2 – Перелік елементів інтерфейсу й бізнес-функцій

№	Елементи інтерфейсу	Бізнес-функція високонавантаженої інформаційної системи
1	Текстове поле «Логін» Текстове поле «Пароль» Кнопка «Вхід»	Використовуються для введення логіна й пароля. Натискання кнопки «Вхід» дозволяє користувачу увійти в систему й змінити свій статус «незареєстрований клієнт» на «User» або «Admin».
2	Кнопка «Оформити замовлення»	Використовується для додавання послуги до списку замовлень.

3	Кнопка «Сплатити»	Використовується для підтвердження друку та подальшої оплати.
4	Чекбокс меню	Використовується для обрання критеріїв сортування
5	Кнопка «Відсортувати»	Використовується для застосування критеріїв сортування
6	Кнопка «Додати макет»	Використовуються для додавання власного макету на друк
7	Кнопка «Прибрати макет»	Використовується для видалення макету в процесі оформлення замовлення
8	Випадаюче меню зміни статусів	Використовується адміністратором для зміни статусу

Таблиця з переліком основних особливостей використання таблиць СУБД MySQL типу MyIsam і InnoDB у високонавантажених базах даних, що визначають їх переваги й недоліки.

Таблиця 1.3 – Порівняльний аналіз таблиць типу MyIsam і InnoDB

№	Опис	MyIsam	InnoDB
1	Підтримка транзакцій – Немає /Так	Немає	Так
2	Підтримка зв'язків за зовнішніми ключами – Немає /Є	Немає	Є
3	Підтримки посилювальної цілісності зв'язків для інструкцій UPDATE, DELETE, INSERT	Немає	Є

4	Блокування – на рівні таблиць / на рівні записів / немає блокування	На рівні таблиць	На рівні записів
5	Одночасні запити до різних записів однієї таблиці – повільніше / швидше.	Повільніше	Швидше
6	Під час змішаного навантаження за запитами SELECT, UPDATE, DELETE, INSERT – повільніше / швидше	Повільніше	Швидше
7	Однотипні операції INSERT – повільніше / швидше	Швидше	Повільніше
8	Однотипні операції SELECT – повільніше / швидше	Швидше	Повільніше
9	Запит Count(*) – повільніше / швидше	Швидше	Повільніше
10	Взаємне блокування операцій (Deadlock) – можлива / неможлива	Неможлива	Можлива
11	Підтримка повнотекстового пошуку – Немає /Є	Є	Немає
12	Підтримка індексування полів у запитах – Немає /Є	Є	Немає
13	Можливі типи індексів	B-tree	B-tree, R-tree
14	Можливість бінарного копіюванні таблиць – Немає /Є	Є	Немає
15	Розмір таблиці БД – укажіть максимальний розмір даних	2 Гб	4 Гб

16	Можливість відновлення у випадку збою – Немає /Є	Немає	Є
17	СУБД створює за замовчуванням для таблиці MyIsam / InnoDB (укажіть кількість файлів, їх розширення, папки для зберігання та їх призначення)	Кожна таблиця зберігається у вигляді двох файлів: файл з даними та файл з індексом. Файли мають розширення .MYD та .MYI відповідно. Дані зберігаються в /var/lib/mysql	База даних зберігається у вигляді одного файлу. Формат файлу залежить від версії MySQL та налаштувань.

**Завдання 1.2.** Розробити серверну частину високонавантаженої інформаційної системи з таблицями СУБД MySQL типу MyIsam, виконавши такий перелік робіт, відповідних до етапів проектування: – створення логічної моделі бази даних у вигляді ER-діаграми згідно з нотацією IDEF1X; – створення бази даних з таблицями типу MyIsam. Для кожної таблиці мають бути розроблені й визначені: SQL-запит, що встановлює тип таблиці; первинний ключ (primary key); індексні поля, зовнішні ключі (foreign key), які надалі використовуватимуться для забезпечення цілісності даних; поля з описом доменів, відповідних до типів даних СУБД MySQL і визначеними атрибутами «NOT NULL» і «NULL». База даних має відповідати вимогам третьої нормальної форми, містити не менш п'яти основних таблиць, з кожною з яких має бути зв'язана хоча б одна підставкова таблиця. Кожна основна таблиця має містити не менш п'яти атрибутів з наступними обов'язковими типами даних: «INT», «VARCHAR», «TEXT», «FLOAT», «DATE»; – створення фізичної моделі бази даних з таблицями типу MyIsam у вигляді ER-діаграм у нотації IDEF1X з



використанням інтерфейсу реверсінжинірингу (reverse engineering) середовища розробки (для програмного пакета Workbench – у вигляді EER-діаграм); – заповнення основних таблиць даними. До основних належать таблиці, що містять дані про клієнтів, товари, послуги тощо.

**Код SQL-запиту, на основі якого створена база даних з таблицями MyIsam:**

```
CREATE TABLE IF NOT EXISTS `mu`.`fonts` (  
  `idFonts` INT NOT NULL AUTO_INCREMENT,  
  `name_font` VARCHAR(45) NOT NULL,  
  `size` INT NOT NULL,  
  `spacing` FLOAT NOT NULL,  
  `color_for_headers` VARCHAR(45) NOT NULL,  
  `color_for_text` VARCHAR(45) NOT NULL,  
  `padding` FLOAT NOT NULL,  
  `style` VARCHAR(45) NOT NULL,  
  `saturation` VARCHAR(45) NOT NULL,  
  `width` INT NOT NULL,  
  `custom_macket` BLOB NOT NULL,  
  `fontsTEXT` TEXT NULL DEFAULT NULL,  
  `fontsDATA` DATE NULL DEFAULT NULL,  
  PRIMARY KEY (`idFonts`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`format_type` (  
  `idFormatType` INT NOT NULL AUTO_INCREMENT,  
  `nameFormat` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idFormatType`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`macket_to_print` (  
  `idMacket_to_Print` INT NOT NULL AUTO_INCREMENT,  
  `SizeByte` INT NOT NULL,  
  `fk_macket_to_formatType` INT NOT NULL,  
  `height` INT NOT NULL,  
  `width` INT NOT NULL,  
  `File` BLOB NOT NULL,  
  `fk_macket_client` INT NOT NULL,
```

```
`individual_specific` TEXT NULL DEFAULT NULL,  
`macketDATA` DATE NULL DEFAULT NULL,  
PRIMARY KEY (`idMacket_to_Print`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`order_status` (  
  `id_order_status` INT NOT NULL AUTO_INCREMENT,  
  `name_status` VARCHAR(45) NOT NULL DEFAULT 'New',  
  PRIMARY KEY (`id_order_status`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`user_role` (  
  `idStatusUser` INT NOT NULL AUTO_INCREMENT,  
  `nameStatus` VARCHAR(100) NOT NULL DEFAULT 'Customer',  
  PRIMARY KEY (`idStatusUser`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`user` (  
  `idUser` INT NOT NULL AUTO_INCREMENT,  
  `nameUser` VARCHAR(45) NOT NULL,  
  `snameUser` VARCHAR(45) NOT NULL,  
  `age` INT NOT NULL,  
  `totalSpent` FLOAT NOT NULL DEFAULT '0',  
  `emailUser` VARCHAR(45) NOT NULL,  
  `phoneNumb` VARCHAR(45) NOT NULL,  
  `gender` TINYINT NULL DEFAULT NULL,  
  `fk_user_to_role` INT NOT NULL,  
  `History` TEXT NULL DEFAULT NULL,  
  `registerDate` DATE NOT NULL,  
  PRIMARY KEY (`idUser`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`order` (  
  `id_order` INT NOT NULL AUTO_INCREMENT,  
  `order_start` DATE NOT NULL,  
  `total_price` FLOAT NOT NULL,  
  `order_done` DATE NOT NULL,
```

```
`orderTEXT` TEXT NOT NULL,  
`orderVARCHAR` VARCHAR(45) NULL DEFAULT '',  
`Details` VARCHAR(45) NULL DEFAULT NULL,  
`fk_order_to_stat` INT NOT NULL,  
`fk_order_user` INT NOT NULL,  
PRIMARY KEY (`id_order`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`order_macket` (  
  `id_order` INT NOT NULL,  
  `id_macket` INT NOT NULL,  
  PRIMARY KEY (`id_order`, `id_macket`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`paper_brightness` (  
  `idPaper_brightness` INT NOT NULL AUTO_INCREMENT,  
  `brighnessLvl` FLOAT NOT NULL,  
  `describe` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`idPaper_brightness`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`paper_density` (  
  `id_density` INT NOT NULL AUTO_INCREMENT,  
  `density_lvl` VARCHAR(45) NOT NULL,  
  `desc` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_density`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`paper` (  
  `idPaper` INT NOT NULL AUTO_INCREMENT,  
  `name_paper` VARCHAR(45) NOT NULL,  
  `iso_format` VARCHAR(5) NOT NULL,  
  `height` INT NOT NULL,  
  `width` INT NOT NULL,  
  `fk_paper_to_bright` INT NOT NULL,  
  `fk_paper_to_density` INT NOT NULL,  
  `thickness` FLOAT NOT NULL,
```

```
`color` VARCHAR(45) NOT NULL,  
`covering` VARCHAR(45) NOT NULL,  
`description` VARCHAR(150) NOT NULL DEFAULT 'not added',  
PRIMARY KEY (`idPaper`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`serivces` (  
  `idSerivces` INT NOT NULL AUTO_INCREMENT,  
  `name_service` VARCHAR(200) NOT NULL,  
  `Description` TEXT NOT NULL,  
  `Price` FLOAT NOT NULL,  
  `code_service` VARCHAR(100) NOT NULL,  
  `fk_service_paper` INT NOT NULL,  
  `serviceDATE` DATE NULL DEFAULT NULL,  
  `fk_service_fonts` INT NOT NULL,  
  PRIMARY KEY (`idSerivces`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `mu`.`order_services` (  
  `fk_order_id` INT NOT NULL,  
  `fk_service_id` INT NOT NULL,  
  `count` INT NOT NULL,  
  PRIMARY KEY (`fk_order_id`, `fk_service_id`))  
ENGINE = MyISAM  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

Скріншот схеми фізичної моделі бази даних з таблицями типу MyIsam у нотації IDEF1X з обов'язковим відображенням первинних і зовнішніх ключів, типів даних, атрибутів «NULL», «NOT NULL», кардинальності зв'язків (для програмного пакета Workbench – у вигляді EER-діаграм, для таблиць – нотація Workbench [Default], для зв'язків – нотація IDEF1X):

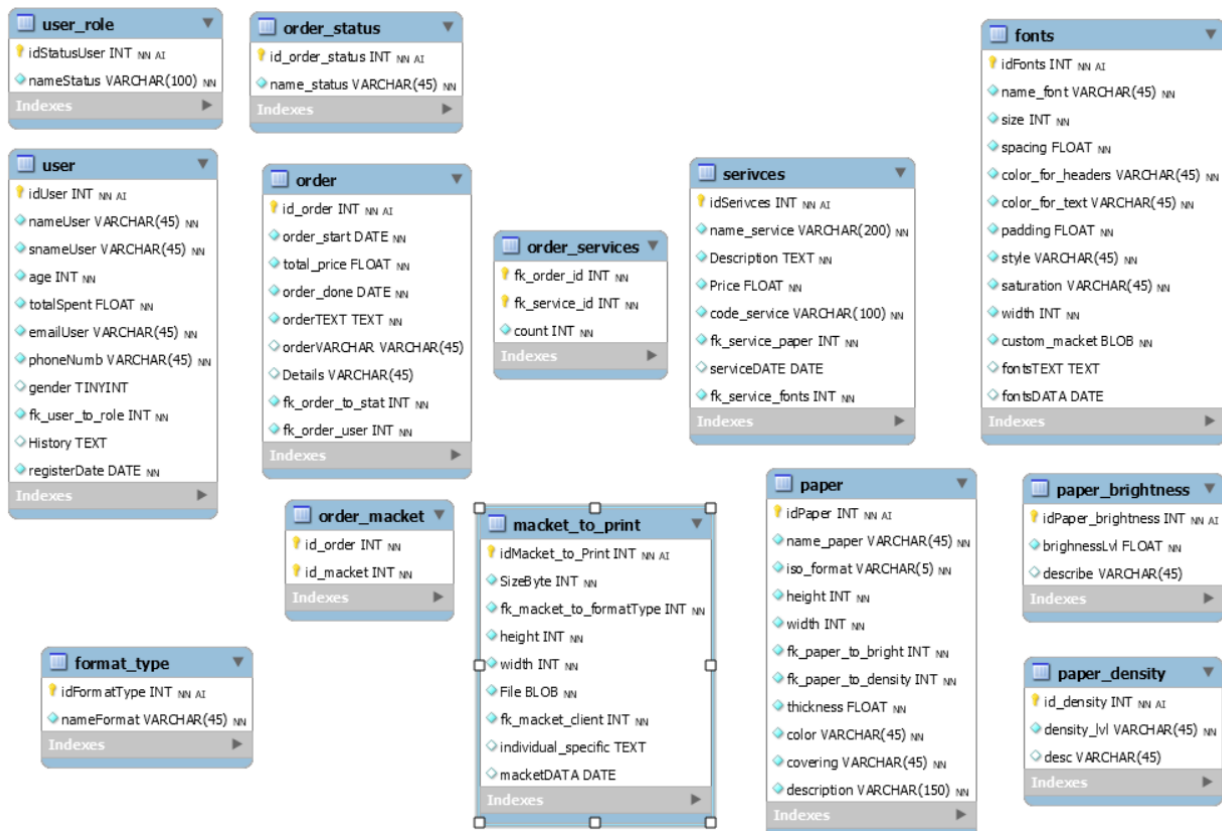


Рисунок 1.1 - Схема фізичної моделі бази даних з таблицями типу MyIsam

**Завдання 1.3.** На основі логічної моделі бази даних завдання 1.2 розробити серверну частину високонавантаженої інформаційної системи з таблицями СУБД MySQL типу InnoDB, виконавши такий перелік робіт, відповідних до етапів проектування: – створення бази даних з таблицями типу InnoDB. Для кожної таблиці мають бути розроблені й визначені: SQL-запит, що встановлює тип таблиці; первинний ключ (primary key); зовнішні ключові поля (foreign key) для забезпечення зв'язку з іншими таблицями; тип посилювальної цілісності зв'язків за зовнішнім ключем (ON UPDATE, CASCADE, SET NULL, NO ACTION, RESTRICT для інструкцій UPDATE і DELETE); поля з описом доменів, відповідних до типів даних СУБД MySQL і визначеними атрибутами «NOT NULL» і «NULL». База даних має повністю відповідати логічній моделі завдання 1.2; – створення фізичної моделі бази даних з таблицями типу InnoDB у вигляді ER-діаграм у нотації IDEF1X з використанням інтерфейсу реверс-інжинірингу (reverse engineering) середовища розробки (для програмного пакета Workbench – у вигляді EER-діаграм); – заповнення основних таблиць даними.

**Код SQL-запиту, на основі якого створена база даних з таблицями InnoDB:**

```
CREATE TABLE IF NOT EXISTS `my`.`user_role` (  
  `idUser` INT NOT NULL AUTO_INCREMENT,  
  `nameStatus` VARCHAR(100) NOT NULL DEFAULT 'Customer',  
  PRIMARY KEY (`idUser`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`user` (  
  `idUser` INT NOT NULL AUTO_INCREMENT,  
  `nameUser` VARCHAR(45) NOT NULL,  
  `snameUser` VARCHAR(45) NOT NULL,  
  `age` INT NOT NULL,  
  `totalSpent` FLOAT NOT NULL DEFAULT '0',  
  `emailUser` VARCHAR(45) NOT NULL,  
  `phoneNumb` VARCHAR(45) NOT NULL,  
  `gender` TINYINT NULL DEFAULT NULL,  
  `fk_user_to_role` INT NOT NULL,  
  `History` TEXT NULL DEFAULT NULL,  
  `registerDate` DATE NOT NULL,  
  PRIMARY KEY (`idUser`),  
  INDEX `fk_status_idx` (`fk_user_to_role` ASC) VISIBLE,  
  CONSTRAINT `fk_user_to_role`  
    FOREIGN KEY (`fk_user_to_role`)  
    REFERENCES `my`.`user_role` (`idUser`)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`order_status` (  
  `id_order_status` INT NOT NULL AUTO_INCREMENT,  
  `name_status` VARCHAR(45) NOT NULL DEFAULT 'New',  
  PRIMARY KEY (`id_order_status`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```

CREATE TABLE IF NOT EXISTS `my`.`order` (
  `id_order` INT NOT NULL AUTO_INCREMENT,
  `order_start` DATE NOT NULL,
  `total_price` FLOAT NOT NULL,
  `order_done` DATE NOT NULL,
  `orderTEXT` TEXT NOT NULL,
  `order` VARCHAR(45) NULL DEFAULT '',
  `Details` VARCHAR(45) NULL DEFAULT NULL,
  `fk_order_to_stat` INT NOT NULL,
  `fk_order_user` INT NOT NULL,
  PRIMARY KEY (`id_order`),
  INDEX `fk_order_to_stat_idx` (`fk_order_to_stat` ASC) VISIBLE,
  INDEX `fk_order_user_idx` (`fk_order_user` ASC) VISIBLE,
  CONSTRAINT `fk_order_to_stat`
    FOREIGN KEY (`fk_order_to_stat`)
      REFERENCES `my`.`order_status` (`id_order_status`)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  CONSTRAINT `fk_order_user`
    FOREIGN KEY (`fk_order_user`)
      REFERENCES `my`.`user` (`idUser`)
      ON DELETE RESTRICT
      ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

```

CREATE TABLE IF NOT EXISTS `my`.`format_type` (
  `idFormatType` INT NOT NULL AUTO_INCREMENT,
  `nameFormat` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idFormatType`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

```

CREATE TABLE IF NOT EXISTS `my`.`macket_to_print` (
  `idMacket_to_Print` INT NOT NULL AUTO_INCREMENT,
  `SizeByte` INT NOT NULL,
  `fk_macket_to_formatType` INT NOT NULL,
  `height` INT NOT NULL,
  `width` INT NOT NULL,
  `File` BLOB NOT NULL,
  `fk_macket_client` INT NOT NULL,

```

```

`individual_specific` TEXT NULL DEFAULT NULL,
`macketDATA` DATE NULL DEFAULT NULL,
PRIMARY KEY (`idMacket_to_Print`),
INDEX `fk_macket_to_formatType_idx` (`fk_macket_to_formatType` ASC)
VISIBLE,
CONSTRAINT `fk_macket_to_formatType`
  FOREIGN KEY (`fk_macket_to_formatType`)
  REFERENCES `my`.`format_type` (`idFormatType`)
  ON DELETE RESTRICT
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

```

CREATE TABLE IF NOT EXISTS `my`.`order_macket` (
  `id_order` INT NOT NULL,
  `id_macket` INT NOT NULL,
  PRIMARY KEY (`id_order`, `id_macket`),
  INDEX `id_macket_idx` (`id_macket` ASC) VISIBLE,
  CONSTRAINT `id_macket`
    FOREIGN KEY (`id_macket`)
    REFERENCES `my`.`macket_to_print` (`idMacket_to_Print`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
  CONSTRAINT `id_order`
    FOREIGN KEY (`id_order`)
    REFERENCES `my`.`order` (`id_order`)
  ON DELETE RESTRICT
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

```

CREATE TABLE IF NOT EXISTS `my`.`paper_density` (
  `id_density` INT NOT NULL AUTO_INCREMENT,
  `density_lvl` VARCHAR(45) NOT NULL,
  `desc` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_density`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

```

CREATE TABLE IF NOT EXISTS `my`.`paper_brightness` (
  `idPaper_brightness` INT NOT NULL AUTO_INCREMENT,
  `brighnessLvl` FLOAT NOT NULL,

```



```
`describe` VARCHAR(45) NULL DEFAULT NULL,  
PRIMARY KEY (`idPaper_brightness`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`paper` (  
  `idPaper` INT NOT NULL AUTO_INCREMENT,  
  `name_paper` VARCHAR(45) NOT NULL,  
  `iso_format` VARCHAR(5) NOT NULL,  
  `height` INT NOT NULL,  
  `width` INT NOT NULL,  
  `fk_paper_to_bright` INT NOT NULL,  
  `fk_paper_to_density` INT NOT NULL,  
  `thickness` FLOAT NOT NULL,  
  `color` VARCHAR(45) NOT NULL,  
  `covering` VARCHAR(45) NOT NULL,  
  `description` VARCHAR(150) NOT NULL DEFAULT 'not added',  
  PRIMARY KEY (`idPaper`),  
  INDEX `fk_paper_to_bright_idx` (`fk_paper_to_bright` ASC) VISIBLE,  
  INDEX `fk_paper_to_density_idx` (`fk_paper_to_density` ASC) VISIBLE,  
  CONSTRAINT `fk_paper_to_bright`  
    FOREIGN KEY (`fk_paper_to_bright`)  
      REFERENCES `my`.`paper_brightness` (`idPaper_brightness`)  
ON DELETE RESTRICT  
ON UPDATE CASCADE,  
  CONSTRAINT `fk_paper_to_density`  
    FOREIGN KEY (`fk_paper_to_density`)  
      REFERENCES `my`.`paper_density` (`id_density`)  
ON DELETE RESTRICT  
ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`fonts` (  
  `idFonts` INT NOT NULL AUTO_INCREMENT,  
  `name_font` VARCHAR(45) NOT NULL,  
  `size` INT NOT NULL,  
  `spacing` FLOAT NOT NULL,  
  `color_for_headers` VARCHAR(45) NOT NULL,  
  `color_for_text` VARCHAR(45) NOT NULL,  
  `padding` FLOAT NOT NULL,  
  `style` VARCHAR(45) NOT NULL,  
  `saturation` VARCHAR(45) NOT NULL,
```

```
`width` INT NOT NULL,  
`custom_maket` BLOB NOT NULL,  
`fontsTEXT` TEXT NULL DEFAULT NULL,  
`fontsDATA` DATE NULL DEFAULT NULL,  
PRIMARY KEY (`idFonts`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`serivces` (  
  `idSerivces` INT NOT NULL AUTO_INCREMENT,  
  `name_service` VARCHAR(200) NOT NULL,  
  `Description` TEXT NOT NULL,  
  `Price` FLOAT NOT NULL,  
  `code_service` VARCHAR(100) NOT NULL,  
  `fk_service_paper` INT NOT NULL,  
  `serviceDATE` DATE NULL DEFAULT NULL,  
  `fk_service_fonts` INT NOT NULL,  
  PRIMARY KEY (`idSerivces`),  
  INDEX `fk_service_paper_idx` (`fk_service_paper` ASC) VISIBLE,  
  INDEX `fk_service_fonts_idx` (`fk_service_fonts` ASC) VISIBLE,  
  CONSTRAINT `fk_service_fonts`  
    FOREIGN KEY (`fk_service_fonts`)  
      REFERENCES `my`.`fonts` (`idFonts`)  
ON DELETE RESTRICT  
ON UPDATE CASCADE,  
  CONSTRAINT `fk_service_paper`  
    FOREIGN KEY (`fk_service_paper`)  
      REFERENCES `my`.`paper` (`idPaper`)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `my`.`order_services` (  
  `fk_order_id` INT NOT NULL,  
  `fk_service_id` INT NOT NULL,  
  `count` INT NOT NULL,  
  PRIMARY KEY (`fk_order_id`, `fk_service_id`),  
  INDEX `fk_service_id_idx` (`fk_service_id` ASC) VISIBLE,  
  CONSTRAINT `fk_order_id`  
    FOREIGN KEY (`fk_order_id`)  
      REFERENCES `my`.`order` (`id_order`)  
    ON DELETE RESTRICT
```

```

ON UPDATE CASCADE,
CONSTRAINT `fk_service_id`
FOREIGN KEY (`fk_service_id`)
REFERENCES `my`.`servces` (`idServces`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

```

Таблиця 1.4 – Перелік типів посилальної цілісності зв'язків за зовнішнім ключем для всіх таблиць типу InnoDB

№	Ім'я таблиці 1, зовнішній ключ	Ім'я таблиці 2, первинний ключ	SQL- інструкція для таблиці 1	Тип посилально ї цілісності
1	fk_user_to_role	idUserUser	UPDATE	CASCADE
			DELETE	RESTRICT
2	fk_order_to_stat	id_order_status	UPDATE	CASCADE
			DELETE	RESTRICT
3	fk_order_user	idUser	UPDATE	CASCADE
			DELETE	RESTRICT
4	fk_macket_to_formatType e	idFormatType	UPDATE	CASCADE
			DELETE	RESTRICT
5	id_macket	idMacket_to_Pri nt	UPDATE	CASCADE
			DELETE	RESTRICT
6	id_order	id_order	UPDATE	CASCADE
			DELETE	RESTRICT
7	fk_paper_to_bright	idPaper_brightne ss	UPDATE	CASCADE
			DELETE	RESTRICT
8	fk_paper_to_density	id_density	UPDATE	CASCADE
			DELETE	RESTRICT
9	fk_service_fonts	idFonts	UPDATE	CASCADE
			DELETE	RESTRICT
10	fk_service_paper	idPaper	UPDATE	CASCADE

			DELETE	RESTRICT
11	fk_order_id	id_order	UPDATE	CASCADE
			DELETE	RESTRICT
12	fk_service_id	idSerivces	UPDATE	CASCADE
			DELETE	RESTRICT

Скріншот схеми фізичної моделі бази даних з таблицями типу InnoDB у нотації IDEF1X з обов'язковим відображенням первинних і зовнішніх ключів, типу даних, атрибутів «NULL», «NOT NULL», кардинальності зв'язків (для програмного пакета Workbench – у вигляді EER-діаграм, для таблиць – нотація Workbench [Default], для зв'язків – нотація IDEF1X):

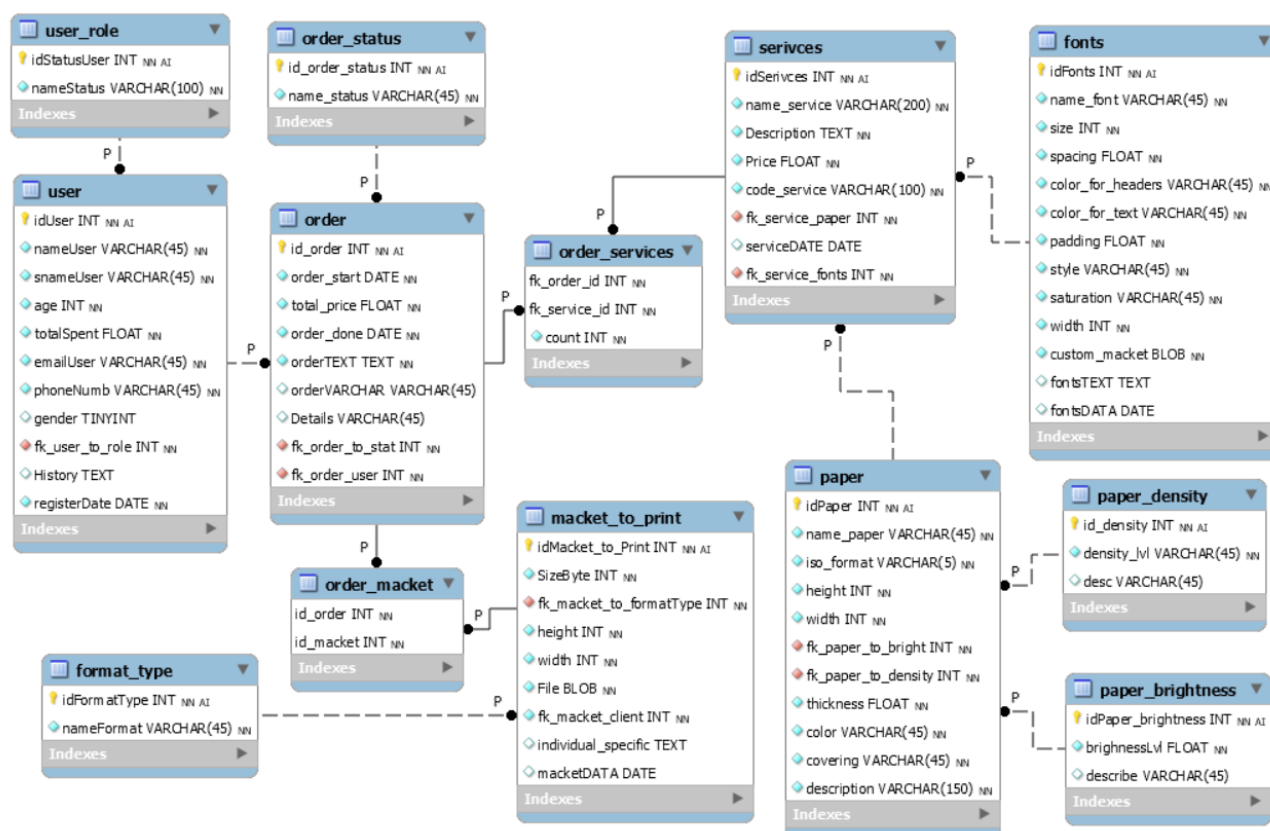


Рисунок 1.2 – Схема фізичної моделі бази даних з таблицями типу InnoDB у нотації IDEF1X

### **3 Висновки**

Виконуючи дане практичне завдання було набуто практичних навичок з розробки баз даних на платформі MySQL з таблицями типу MyISAM і InnoDB, з урахуванням особливостей роботи високонавантаженої інформаційної системи зберігання даних.

Після порівняння InnoDB та MyISAM можна зробити такі висновки:

1. Краще використовувати MyISAM в таблицях, де переважають один тип доступу: читання (сайт новин) або написання (наприклад, журналювання) ;
2. Використання InnoDB має сенс у всіх інших випадках і випадках підвищених вимог безпеки даних, через можливість відновлення завдяки лог-файлам.