МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра "Комп'ютеризовані системи автоматики"

Звіт

з лабораторної роботи № 3

«Створення моделі бази даних»
з навчальної дисципліни: «Організація баз даних та знань»

Виконала:

гр. ІР-22

Анастасія Кобрин

Прийняла:
к.т.н., доц
Лагун Ілона Ігорівна

Львів
2025

Завдання

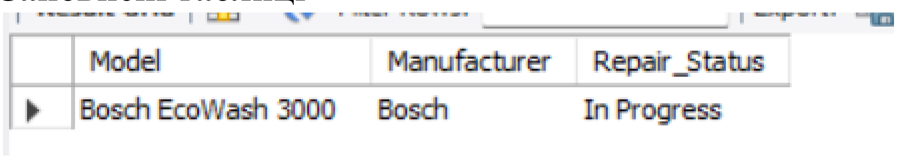
Завдання виконується на основі спроектованої моделі БД в лаб.роботі №1.
Для даної моделі слід здійснити повторний аналіз структури БД на предмет забезпечення 3ї нормальної форми, наявності усіх Primary Key.
Для реалізації БД написати сценарій для її створення у ядрі MySQL, який повинен відповідати наступним критеріям:

❖ SQL-скріпт можна запускати декілька раз підряд і при цьому не повинні з'являтися помилки через існуючі елементи БД. Тобто повинні бути передбачені відповідні DROP для таблиць та IF EXISTS на створення БД.

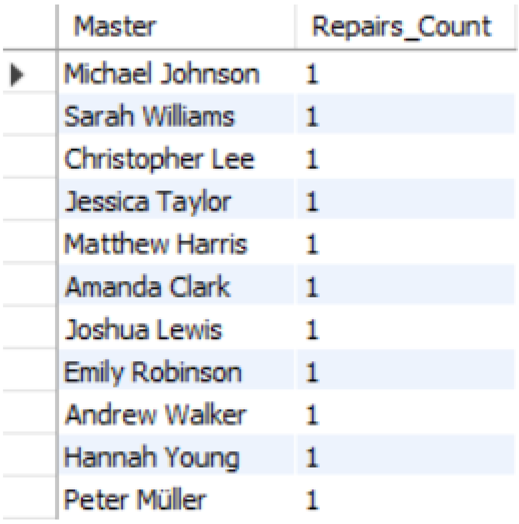
❖ Для кожної таблиці слід здійснити заповнення даними по 10-15 INSERT, при цьому передбачити правильний порядок вставки даних.

❖ Створити для таблиць як мінімум 2 додаткові індекси, згідно виконаного попереднього аналізу структури даних.

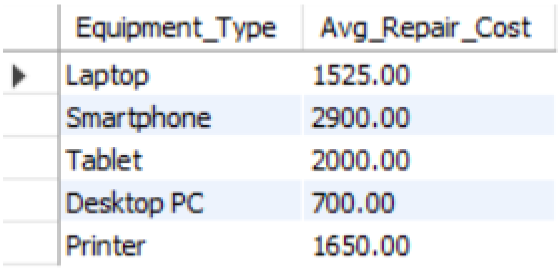
Заповнені таблиці



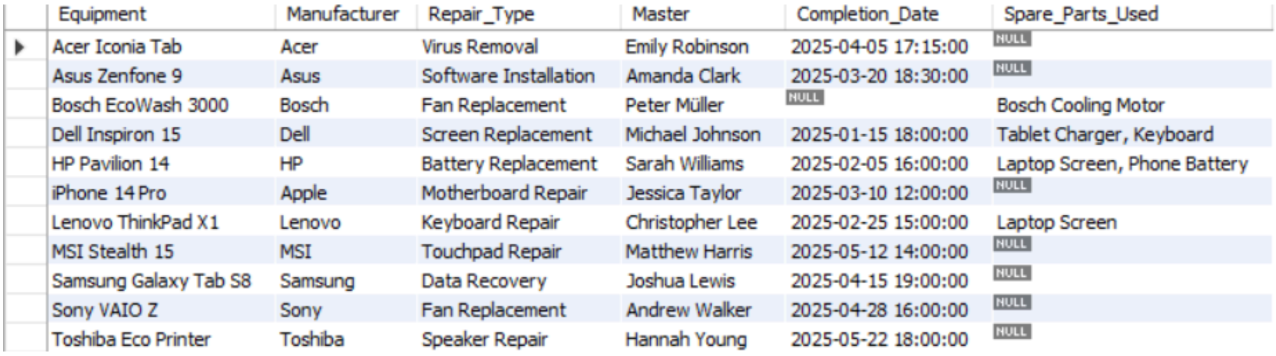
Model	Manufacturer	Repair_Status
Bosch EcoWash 3000	Bosch	In Progress



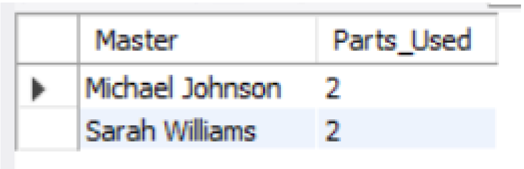
Master	Repairs_Count
Michael Johnson	1
Sarah Williams	1
Christopher Lee	1
Jessica Taylor	1
Matthew Harris	1
Amanda Clark	1
Joshua Lewis	1
Emily Robinson	1
Andrew Walker	1
Hannah Young	1
Peter Müller	1



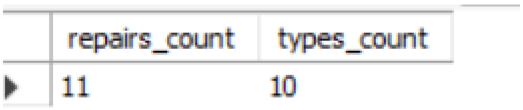
Equipment_Type	Avg_Repair_Cost
Laptop	1525.00
Smartphone	2900.00
Tablet	2000.00
Desktop PC	700.00
Printer	1650.00



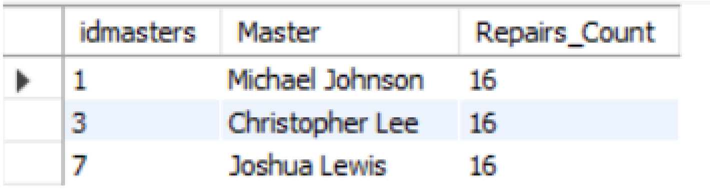
Equipment	Manufacturer	Repair_Type	Master	Completion_Date	Spare_Parts_Used
Acer Iconia Tab	Acer	Virus Removal	Emily Robinson	2025-04-05 17:15:00	NULL
Asus Zenfone 9	Asus	Software Installation	Amanda Clark	2025-03-20 18:30:00	NULL
Bosch EcoWash 3000	Bosch	Fan Replacement	Peter Müller	NULL	Bosch Cooling Motor
Dell Inspiron 15	Dell	Screen Replacement	Michael Johnson	2025-01-15 18:00:00	Tablet Charger, Keyboard
HP Pavilion 14	HP	Battery Replacement	Sarah Williams	2025-02-05 16:00:00	Laptop Screen, Phone Battery
iPhone 14 Pro	Apple	Motherboard Repair	Jessica Taylor	2025-03-10 12:00:00	NULL
Lenovo ThinkPad X1	Lenovo	Keyboard Repair	Christopher Lee	2025-02-25 15:00:00	Laptop Screen
MSI Stealth 15	MSI	Touchpad Repair	Matthew Harris	2025-05-12 14:00:00	NULL
Samsung Galaxy Tab S8	Samsung	Data Recovery	Joshua Lewis	2025-04-15 19:00:00	NULL
Sony VAIO Z	Sony	Fan Replacement	Andrew Walker	2025-04-28 16:00:00	NULL
Toshiba Eco Printer	Toshiba	Speaker Repair	Hannah Young	2025-05-22 18:00:00	NULL



Master	Parts_Used
Michael Johnson	2
Sarah Williams	2



repairs_count	types_count
11	10



idmasters	Master	Repairs_Count
1	Michael Johnson	16
3	Christopher Lee	16
7	Joshua Lewis	16

Код

```
CREATE DATABASE IF NOT EXISTS `mydb`;
USE `mydb`;

DROP TABLE IF EXISTS repair_job_spare_part;
DROP TABLE IF EXISTS masters_schedule;
DROP TABLE IF EXISTS spare_parts;
DROP TABLE IF EXISTS repair_jobs;
DROP TABLE IF EXISTS repairs;
DROP TABLE IF EXISTS equipment;
DROP TABLE IF EXISTS clients;
DROP TABLE IF EXISTS masters;
DROP TABLE IF EXISTS repair_types;
DROP TABLE IF EXISTS equipment_types;
DROP TABLE IF EXISTS manufacturers;


CREATE TABLE IF NOT EXISTS `mydb`.`manufacturers` (
  `idmanufacturers` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `country` VARCHAR(45) NULL,
  `website` VARCHAR(45) NULL,
  `phone` VARCHAR(45) NULL,
  PRIMARY KEY (`idmanufacturers`),
  UNIQUE INDEX `phone_UNIQUE` (`phone` ASC) VISIBLE
  );


INSERT INTO manufacturers (idmanufacturers, name, country, website, phone)
VALUES
```

```
(1, 'Dell', 'USA', 'www.dell.com', '+18005551234'),
(2, 'HP', 'USA', 'www.hp.com', '+18005554321'),
(3, 'Lenovo', 'China', 'www.lenovo.com', '+8613800138000'),
(4, 'Apple', 'USA', 'www.apple.com', '+18006927753'),
(5, 'Asus', 'Taiwan', 'www.asus.com', '+886223456789'),
(6, 'Acer', 'Taiwan', 'www.acer.com', '+886223498888'),
(7, 'Samsung', 'South Korea', 'www.samsung.com', '+82234567890'),
(8, 'Sony', 'Japan', 'www.sony.com', '+81334567890'),
(9, 'MSI', 'Taiwan', 'www.msi.com', '+886223499999'),
(10, 'Toshiba', 'Japan', 'www.toshiba.com', '+81312345678'),
(11, 'Bosch', 'Germany', 'www.bosch.com', '+491234567890');


CREATE TABLE IF NOT EXISTS `mydb`.`equipment_types` (
  `idequipment_types` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `description` TEXT(200) NULL,
  PRIMARY KEY (`idequipment_types`));

INSERT INTO equipment_types (name, description) VALUES
('Laptop', 'Portable personal computer'),
('Smartphone', 'Mobile communication device'),
('Tablet', 'Touchscreen portable device'),
('Desktop PC', 'Stationary personal computer'),
('Printer', 'Device for printing documents'),
('Monitor', 'Display device'),
('Server', 'High-performance computer for networks'),
('Smartwatch', 'Wearable smart device'),
('Router', 'Network routing device'),
('Camera', 'Digital photo/video device');

CREATE TABLE IF NOT EXISTS `mydb`.`equipment` (
  `idequipment` INT NOT NULL AUTO_INCREMENT,
  `serial_number` INT NULL,
  `manufacturer_id` INT NULL,
  `equipment_type_id` INT NULL,
  `model` VARCHAR(45) NULL,
  `warranty_until` DATE NULL,
  PRIMARY KEY (`idequipment`),
  INDEX `fk_equipment_manufacturer_idx` (`manufacturer_id` ASC) VISIBLE,
  INDEX `fk_equipment_type_idx` (`equipment_type_id` ASC) VISIBLE,
  UNIQUE INDEX `index_serial_number` (`serial_number` ASC) VISIBLE);

  alter table equipment
        add CONSTRAINT `fk_equipment_manufacturer`
```

```
    FOREIGN KEY (`manufacturer_id`)
    REFERENCES `mydb`.`manufacturers` (`idmanufacturers`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

        add CONSTRAINT `fk_equipment_type`
    FOREIGN KEY (`equipment_type_id`)
    REFERENCES `mydb`.`equipment_types` (`idequipment_types`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;


INSERT INTO equipment (serial_number, manufacturer_id, equipment_type_id,
model, warranty_until) VALUES
(10001, 1, 1, 'Dell Inspiron 15', '2026-05-10'),
(10002, 2, 1, 'HP Pavilion 14', '2025-12-20'),
(10003, 3, 1, 'Lenovo ThinkPad X1', '2026-03-15'),
(10004, 4, 2, 'iPhone 14 Pro', '2026-09-01'),
(10005, 5, 2, 'Asus Zenfone 9', '2025-11-05'),
(10006, 6, 3, 'Acer Iconia Tab', '2025-07-30'),
(10007, 7, 3, 'Samsung Galaxy Tab S8', '2026-02-28'),
(10008, 8, 4, 'Sony VAIO Z', '2025-10-12'),
(10009, 9, 1, 'MSI Stealth 15', '2026-01-22'),
(10010, 10, 5, 'Toshiba Eco Printer', '2025-08-18'),
(10011, (SELECT idmanufacturers FROM manufacturers WHERE name = 'Bosch'),
        5, 'Bosch EcoWash 3000', '2026-12-31');


CREATE TABLE IF NOT EXISTS `mydb`.`clients` (
  `idclients` INT NOT NULL AUTO_INCREMENT,
  `full_name` VARCHAR(45) NULL,
  `phone` VARCHAR(45) NULL,
  `email` VARCHAR(45) NULL,
  `registration_date` DATE NULL,
  PRIMARY KEY (`idclients`),
  UNIQUE INDEX `phone_UNIQUE` (`phone` ASC) VISIBLE);

INSERT INTO clients (full_name, phone, email, registration_date) VALUES
('John Smith', '+123456789', 'john.smith@email.com', '2024-01-10'),
('Alice Johnson', '+987654321', 'alice.j@email.com', '2024-02-15'),
('Robert Brown', '+192837465', 'robert.b@email.com', '2024-03-20'),
('Maria Garcia', '+564738291', 'maria.g@email.com', '2024-04-05'),
('David Wilson', '+135792468', 'david.w@email.com', '2024-05-01'),
('Emma Davis', '+246813579', 'emma.d@email.com', '2024-05-12'),
('Daniel Miller', '+111222333', 'daniel.m@email.com', '2024-06-01'),
('Sophia Martinez', '+444555666', 'sophia.m@email.com', '2024-06-18'),
('James Anderson', '+777888999', 'james.a@email.com', '2024-07-02'),
('Olivia Thomas', '+101202303', 'olivia.t@email.com', '2024-07-15'),
('Laura Schmidt', '+490111222333', 'laura.s@email.com', '2025-02-10');


CREATE TABLE IF NOT EXISTS `mydb`.`repairs` (
  `idrepairs` INT NOT NULL AUTO_INCREMENT,
  `equipment_id` INT NULL,
  `client_id` INT NULL,
  `start_date` DATETIME NULL,
  `end_date` DATETIME NULL,
  `status` VARCHAR(45) NULL,
  PRIMARY KEY (`idrepairs`),
  INDEX `fk_repairs_equipment_idx` (`equipment_id` ASC) VISIBLE,
  INDEX `fk_repairs_client_idx` (`client_id` ASC) VISIBLE);

  alter table repairs
        add CONSTRAINT `fk_repairs_equipment`
    FOREIGN KEY (`equipment_id`)
    REFERENCES `mydb`.`equipment` (`idequipment`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

        add CONSTRAINT `fk_repairs_client`
    FOREIGN KEY (`client_id`)
    REFERENCES `mydb`.`clients` (`idclients`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;


INSERT INTO repairs (equipment_id, client_id, start_date, end_date, status)
VALUES
(1, 1, '2025-01-10 09:00:00', '2025-01-15 18:00:00', 'Completed'),
(2, 2, '2025-02-01 10:00:00', '2025-02-05 16:00:00', 'Completed'),
(3, 3, '2025-02-20 11:00:00', '2025-02-25 15:00:00', 'Completed'),
(4, 4, '2025-03-01 14:00:00', '2025-03-10 12:00:00', 'In Progress'),
(5, 5, '2025-03-15 09:30:00', '2025-03-20 18:30:00', 'Completed'),
(6, 6, '2025-04-01 08:45:00', '2025-04-05 17:15:00', 'Completed'),
(7, 7, '2025-04-10 10:00:00', '2025-04-15 19:00:00', 'Completed'),
(8, 8, '2025-04-20 13:00:00', '2025-04-28 16:00:00', 'In Progress'),
(9, 9, '2025-05-05 09:00:00', '2025-05-12 14:00:00', 'Completed'),
(10, 10, '2025-05-15 10:00:00', '2025-05-22 18:00:00', 'Completed'),
((SELECT idequipment FROM equipment WHERE model = 'Bosch EcoWash
3000'),
(SELECT idclients FROM clients WHERE full_name = 'Laura Schmidt'),
'2025-10-01 09:00:00', NULL, 'In Progress');


CREATE TABLE IF NOT EXISTS `mydb`.`repair_types` (
  `idrepair_types` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `description` VARCHAR(80) NULL,
  PRIMARY KEY (`idrepair_types`));

INSERT INTO repair_types (name, description) VALUES
('Screen Replacement', 'Replace broken display'),
('Battery Replacement', 'Replace old battery'),
('Keyboard Repair', 'Fix or replace keyboard'),
('Motherboard Repair', 'Repair motherboard issues'),
('Software Installation', 'Install or update software'),
('Virus Removal', 'Remove malware and viruses'),
('Data Recovery', 'Restore lost files'),
('Fan Replacement', 'Replace cooling fan'),
('Touchpad Repair', 'Fix touchpad issues'),
('Speaker Repair', 'Fix sound issues');


CREATE TABLE IF NOT EXISTS `mydb`.`masters` (
  `idmasters` INT NOT NULL AUTO_INCREMENT,
  `full_name` VARCHAR(45) NULL,
  `specialization` VARCHAR(45) NULL,
  `phone` VARCHAR(45) NULL,
  PRIMARY KEY (`idmasters`),
  UNIQUE INDEX `phone_UNIQUE` (`phone` ASC) VISIBLE);

INSERT INTO masters (full_name, specialization, phone) VALUES
('Michael Johnson', 'Laptop Repair', '+100200300'),
('Sarah Williams', 'Smartphone Repair', '+200300400'),
('Christopher Lee', 'Tablet Repair', '+300400500'),
('Jessica Taylor', 'PC Repair', '+400500600'),
('Matthew Harris', 'Printer Repair', '+500600700'),
('Amanda Clark', 'Software Specialist', '+600700800'),
('Joshua Lewis', 'Network Specialist', '+700800900'),
('Emily Robinson', 'Electronics Repair', '+800900100'),
('Andrew Walker', 'Motherboard Repair', '+900100200'),
('Hannah Young', 'General Technician', '+101202303'),
('Peter Müller', 'Appliance Repair', '+490555666777');
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`repair_jobs` (
  `idrepair_jobs` INT NOT NULL AUTO_INCREMENT,
  `repair_id` INT NULL,
  `repair_type_id` INT NULL,
  `cost` INT NULL,
  `master_id` INT NULL,
  PRIMARY KEY (`idrepair_jobs`),
  INDEX `fk_jobs_repairs_idx` (`repair_id` ASC) VISIBLE,
  INDEX `fk_jobs_type_idx` (`repair_type_id` ASC) VISIBLE,
  INDEX `fk_jobs_masters_idx` (`master_id` ASC) VISIBLE);

  alter table repair_jobs
        add CONSTRAINT `fk_jobs_repairs`
    FOREIGN KEY (`repair_id`)
    REFERENCES `mydb`.`repairs` (`idrepairs`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

        add CONSTRAINT `fk_jobs_type`
    FOREIGN KEY (`repair_type_id`)
    REFERENCES `mydb`.`repair_types` (`idrepair_types`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
        add CONSTRAINT `fk_jobs_masters`
    FOREIGN KEY (`master_id`)
    REFERENCES `mydb`.`masters` (`idmasters`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;


INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id) VALUES
(1, 1, 2500, 1),
(2, 2, 1200, 2),
(3, 3, 1500, 3),
(4, 4, 5000, 4),
(5, 5, 800, 6),
(6, 6, 1000, 8),
(7, 7, 3000, 7),
(8, 8, 700, 9),
(9, 9, 900, 5),
(10, 10, 1100, 10),
((SELECT idrepairs FROM repairs r JOIN equipment e ON r.equipment_id =
e.idequipment
WHERE e.model = 'Bosch EcoWash 3000'),
(SELECT idrepair_types FROM repair_types WHERE name = 'Fan
Replacement'),2200,
(SELECT idmasters FROM masters WHERE full_name = 'Peter Müller'));

CREATE TABLE IF NOT EXISTS `mydb`.`spare_parts` (
  `idspare_parts` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `manufacturer_id` INT NULL,
  `equipment_type_id` INT NULL,
  `price` INT NULL,
  `stock_quantity` INT NULL,
  PRIMARY KEY (`idspare_parts`),
  INDEX `fk_spare_manufacturer_idx` (`manufacturer_id` ASC) VISIBLE,
  INDEX `fki_spare_type_idx` (`equipment_type_id` ASC) VISIBLE,
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE);

  alter table spare_parts
	add CONSTRAINT `fk_spare_manufacturer`
    FOREIGN KEY (`manufacturer_id`)
    REFERENCES `mydb`.`manufacturers` (`idmanufacturers`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

	add CONSTRAINT `fk_spare_type`
    FOREIGN KEY (`equipment_type_id`)
    REFERENCES `mydb`.`equipment_types` (`idequipment_types`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;


INSERT INTO spare_parts (name, manufacturer_id, equipment_type_id, price,
stock_quantity) VALUES
('Laptop Screen', 1, 1, 2000, 10),
('Phone Battery', 2, 2, 800, 20),
('Tablet Charger', 3, 3, 600, 15),
('Keyboard', 1, 1, 500, 25),
('Motherboard', 2, 1, 4000, 5),
('SSD 512GB', 3, 1, 3500, 8),
('RAM 16GB', 1, 1, 2500, 12),
('Cooling Fan', 2, 1, 700, 18),
('Power Adapter', 3, 2, 900, 10),
('Touchpad', 1, 1, 1200, 6),
 ('Bosch Cooling Motor', (SELECT idmanufacturers FROM manufacturers WHERE
name = 'Bosch'), 5, 1800, 7);
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`repair_job_spare_part` (
 `repair_job_id` INT NOT NULL,
  `spare_part_id` INT NOT NULL,
  `quantity` INT NOT NULL,
  PRIMARY KEY (`repair_job_id`, `spare_part_id`));

  alter table repair_job_spare_part
        add CONSTRAINT `fk_rjsp_jobs`
    FOREIGN KEY (`repair_job_id`)
    REFERENCES `mydb`.`repair_jobs` (`idrepair_jobs`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,

        add CONSTRAINT `fk_rjsp_spare`
    FOREIGN KEY (`spare_part_id`)
    REFERENCES `mydb`.`spare_parts` (`idspare_parts`)
    ON DELETE CASCADE
    ON UPDATE CASCADE;


INSERT INTO repair_job_spare_part (repair_job_id, spare_part_id, quantity)
VALUES
(1, 3, 1),
(1, 4, 1),
(2, 1, 1),
(2, 2, 2),
(3, 1, 1),
((SELECT idrepair_jobs FROM repair_jobs rj JOIN repairs r ON rj.repair_id =
r.idrepairs
JOIN equipment e ON r.equipment_id = e.idequipment
WHERE e.model = 'Bosch EcoWash 3000'), (SELECT idspare_parts FROM
spare_parts WHERE name = 'Bosch Cooling Motor'),1 );

CREATE TABLE IF NOT EXISTS `mydb`.`masters_schedule` (
  `idmasters_schedule` INT NOT NULL AUTO_INCREMENT,
  `masters_id` INT NULL,
  `work_date` DATETIME NULL,
  `shift` VARCHAR(45) NULL,
  PRIMARY KEY (`idmasters_schedule`),
  INDEX `fk_schedule_masters_idx` (`masters_id` ASC) VISIBLE)
  ENGINE = InnoDB;
  alter table masters_schedule
        add CONSTRAINT `fk_schedule_masters`
    FOREIGN KEY (`masters_id`)
    REFERENCES `mydb`.`masters` (`idmasters`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;


INSERT INTO masters_schedule (masters_id, work_date, shift) VALUES
(1, '2025-05-01 09:00:00', 'Morning'),
(2, '2025-05-01 14:00:00', 'Evening'),
(3, '2025-05-02 09:00:00', 'Morning'),
(4, '2025-05-02 14:00:00', 'Evening'),
(5, '2025-05-03 09:00:00', 'Morning'),
(6, '2025-05-03 14:00:00', 'Evening'),
(7, '2025-05-04 09:00:00', 'Morning'),
(8, '2025-05-04 14:00:00', 'Evening'),
(9, '2025-05-05 09:00:00', 'Morning'),
(10, '2025-05-05 14:00:00', 'Evening');


SELECT e.model AS Model, m.name AS Manufacturer, r.status AS Repair_Status
FROM equipment e
JOIN manufacturers m ON e.manufacturer_id = m.idmanufacturers
JOIN repairs r ON r.equipment_id = e.idequipment
WHERE m.name = 'Bosch' AND r.status = 'In Progress';


SELECT et.name AS Equipment_Type, ROUND(AVG(rj.cost), 2) AS
Avg_Repair_Cost FROM repair_jobs rj
JOIN repairs r ON rj.repair_id = r.idrepairs
JOIN equipment e ON r.equipment_id = e.idequipment
JOIN equipment_types et ON e.equipment_type_id = et.idequipment_types
GROUP BY et.name;


SELECT m.full_name AS Master, COUNT(rj.idrepair_jobs) AS Repairs_Count
FROM repair_jobs rj
JOIN masters m ON rj.master_id = m.idmasters
GROUP BY m.full_name
HAVING COUNT(rj.idrepair_jobs) >= 1;


SELECT
    e.model AS Equipment,
    man.name AS Manufacturer,
    rt.name AS Repair_Type,
    ms.full_name AS Master,
    r.end_date AS Completion_Date,
    GROUP_CONCAT(sp.name SEPARATOR ', ') AS Spare_Parts_Used
FROM repairs r
JOIN equipment e ON r.equipment_id = e.idequipment
JOIN manufacturers man ON e.manufacturer_id = man.idmanufacturers
JOIN repair_jobs rj ON rj.repair_id = r.idrepairs
JOIN repair_types rt ON rj.repair_type_id = rt.idrepair_types
JOIN masters ms ON rj.master_id = ms.idmasters
LEFT JOIN repair_job_spare_part rjsp ON rjsp.repair_job_id = rj.idrepair_jobs
LEFT JOIN spare_parts sp ON rjsp.spare_part_id = sp.idspare_parts
GROUP BY e.model, man.name, rt.name, ms.full_name, r.end_date;


SELECT m.full_name AS Master, COUNT(rjsp.spare_part_id) AS Parts_Used
FROM masters m
JOIN repair_jobs rj ON m.idmasters = rj.master_id
JOIN repair_job_spare_part rjsp ON rj.idrepair_jobs = rjsp.repair_job_id
GROUP BY m.full_name
HAVING COUNT(rjsp.spare_part_id) > (
    SELECT AVG(part_count)
    FROM (
        SELECT COUNT(rjsp2.spare_part_id) AS part_count
        FROM repair_job_spare_part rjsp2
        GROUP BY rjsp2.repair_job_id
    ) AS avg_parts
);

DROP TEMPORARY TABLE IF EXISTS seq;
CREATE TEMPORARY TABLE seq (n INT NOT NULL PRIMARY KEY);
INSERT INTO seq (n) VALUES
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10),
(11),(12),(13),(14),(15),(16),(17),(18),(19),(20);

SET @repairs_cnt = (SELECT COUNT(*) FROM repairs);
SET @types_cnt   = (SELECT COUNT(*) FROM repair_types);

SELECT @repairs_cnt AS repairs_count, @types_cnt AS types_count;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT
    ((n-1) % @repairs_cnt) + 1 AS repair_id,
    FLOOR(1 + RAND() * @types_cnt) AS repair_type_id,
    FLOOR(500 + RAND() * 3000) AS cost,
    1
FROM seq;
```

```
INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT
   ((n-1) % @repairs_cnt) + 1,
   FLOOR(1 + RAND() * @types_cnt),
   FLOOR(500 + RAND() * 3000),
   3
FROM seq;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT
   ((n-1) % @repairs_cnt) + 1,
   FLOOR(1 + RAND() * @types_cnt),
   FLOOR(500 + RAND() * 3000),
   7
FROM seq;


INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 2
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 4
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 5
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 6
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 8
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 9
```

```
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 10
FROM seq WHERE n <= 3;

INSERT INTO repair_jobs (repair_id, repair_type_id, cost, master_id)
SELECT ((n-1) % @repairs_cnt) + 1, FLOOR(1 + RAND() * @types_cnt),
FLOOR(500 + RAND() * 3000), 11
FROM seq WHERE n <= 3;

SELECT
    m.idmasters,
    m.full_name AS Master,
    COUNT(rj.idrepair_jobs) AS Repairs_Count
FROM repair_jobs rj
JOIN masters m ON rj.master_id = m.idmasters
JOIN repairs r ON rj.repair_id = r.idrepairs
WHERE r.status = 'Completed' OR r.status IS NULL
GROUP BY m.idmasters, m.full_name
HAVING COUNT(rj.idrepair_jobs) > 15
ORDER BY Repairs_Count DESC;

DROP TEMPORARY TABLE IF EXISTS seq;

CREATE INDEX idx_repairs_status ON repairs (status);

CREATE INDEX idx_equipment_manufacturer_type ON equipment
(manufacturer_id, equipment_type_id);

CREATE INDEX idx_equipment_types_name ON equipment_types (name);

CREATE INDEX idx_clients_email ON clients (email);

CREATE INDEX idx_repairs_client_equipment ON repairs (client_id,
equipment_id);

CREATE INDEX idx_spare_parts_price ON spare_parts (price);

CREATE INDEX idx_masters_schedule_work_date ON masters_schedule
(work_date);

CREATE INDEX idx_repair_jobs_cost_type ON repair_jobs (repair_type_id, cost);
```