8 ЗАПИТИ ДО БАЗИ ДАНИХ

Код створення бази даних та додавання таблиць наступний:

CREATE DATABASE IF NOT EXISTS `PhotocopyPoint`;

USE `PhotocopyPoint`;

CREATE TABLE IF NOT EXISTS `Contact\_info` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`full\_name` VARCHAR(255) NOT NULL,

`phone\_number` VARCHAR(14) NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE IF NOT EXISTS `PickUpStation` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`address` VARCHAR(255) NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE IF NOT EXISTS `Money` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`count` DECIMAL(19,4) NOT NULL,

`unit` VARCHAR(50) NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE IF NOT EXISTS `Price` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`money\_id` BIGINT UNSIGNED NOT NULL,

`create\_time` DATETIME NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`money\_id`) REFERENCES `Money`(`id`)

);

CREATE TABLE IF NOT EXISTS `TypeService` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(255) NOT NULL,

`info` VARCHAR(255) NOT NULL,

`term` TIME NOT NULL,

`price\_id` BIGINT UNSIGNED NOT NULL,

`create\_time` DATETIME NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`price\_id`) REFERENCES `Price`(`id`)

);

CREATE TABLE IF NOT EXISTS `Client` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`station\_name` VARCHAR(255) NOT NULL,

`contact\_info\_id` BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`contact\_info\_id`) REFERENCES `Contact\_info`(`id`)

);

CREATE TABLE IF NOT EXISTS `Staff` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`address` VARCHAR(255) NOT NULL,

`contact\_info\_id` BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`contact\_info\_id`) REFERENCES `Contact\_info`(`id`)

);

CREATE TABLE IF NOT EXISTS `Order` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`money\_id` BIGINT UNSIGNED NOT NULL,

`client\_id` BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`money\_id`) REFERENCES `Money`(`id`),

FOREIGN KEY (`client\_id`) REFERENCES `Client`(`id`)

);

CREATE TABLE IF NOT EXISTS `Ticket` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`order\_id` BIGINT UNSIGNED NOT NULL,

`isUsed` BOOLEAN NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`order\_id`) REFERENCES `Order`(`id`)

);

CREATE TABLE IF NOT EXISTS `Message` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`text` VARCHAR(1000) NOT NULL,

`create\_time` DATETIME NOT NULL,

`order\_id` BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`order\_id`) REFERENCES `Order`(`id`)

);

CREATE TABLE IF NOT EXISTS `TypeServiceOrder` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`order\_id` BIGINT UNSIGNED NOT NULL,

`term` DATETIME NOT NULL,

`type\_service\_id` BIGINT UNSIGNED NOT NULL,

`count` INT NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`order\_id`) REFERENCES `Order`(`id`),

FOREIGN KEY (`type\_service\_id`) REFERENCES `TypeService`(`id`)

);

CREATE TABLE IF NOT EXISTS `OrderStaff` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`order\_id` BIGINT UNSIGNED NOT NULL,

`staff\_id` BIGINT UNSIGNED NOT NULL,

`specialization` VARCHAR(255) NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`order\_id`) REFERENCES `Order`(`id`),

FOREIGN KEY (`staff\_id`) REFERENCES `Staff`(`id`)

);

CREATE TABLE IF NOT EXISTS `PickUpStationStaff` (

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`staff\_id` BIGINT UNSIGNED,

`pickUpStation\_id` BIGINT UNSIGNED,

PRIMARY KEY (`id`),

FOREIGN KEY (`staff\_id`) REFERENCES `Staff`(`id`),

FOREIGN KEY (`pickUpStation\_id`) REFERENCES `PickUpStation`(`id`)

);

Додавання даних.

INSERT INTO `Contact\_info` (`full\_name`, `phone\_number`) VALUES

('Іван Іванов', '+380501234567'),

('Петро Петренко', '+380671234568');

INSERT INTO `PickUpStation` (`address`) VALUES

('Київ, вул. Хрещатик, 1'),

('Львів, просп. Свободи, 45');

INSERT INTO `Money` (`count`, `unit`) VALUES

(100.00, 'UAH'),

(200.00, 'UAH');

INSERT INTO `Price` (`money\_id`, `create\_time`) VALUES

(1, NOW()),

(2, NOW());

INSERT INTO `TypeService` (`name`, `info`, `term`, `price\_id`, `create\_time`) VALUES

('Друк', 'Чорно-білий друк', '00:01:00', 1, NOW()),

('Копіювання', 'Кольорове копіювання', '00:02:30', 2, NOW());

INSERT INTO `Client` (`station\_name`, `contact\_info\_id`) VALUES

('Київська станція', 1);

INSERT INTO `Staff` (`address`, `contact\_info\_id`) VALUES

('Київ, вул. Саксаганського, 12', 2);

INSERT INTO `Order` (`money\_id`, `client\_id`) VALUES

(1, 1);

INSERT INTO `Ticket` (`order\_id`, `isUsed`) VALUES

(1, FALSE);

INSERT INTO `Message` (`text`, `create\_time`, `order\_id`) VALUES

('Повідомлення для оператора принтера: додаткова інформація', NOW(), 1);

INSERT INTO `TypeServiceOrder` (`term`, `order\_id`, `type\_service\_id`, `count`) VALUES

(NOW() + INTERVAL 1 MINUTE, 1, 1, 10);

INSERT INTO `OrderStaff` (`staff\_id`, `order\_id`, `specialization`) VALUES

(1, 1, 'Оператор принтера');

INSERT INTO `PickUpStationStaff` (`staff\_id`, `pickUpStation\_id`) VALUES

(1, 1);

Отримати усі типи послуг та їх ціну за новою редакцією та назвою яка починається на «д»

SELECT TS.name,

TS.id,

TS.info,

TS.term,

PM.count,

PM.unit,

TS.create\_time

FROM PhotocopyPoint.TypeService TS

INNER JOIN (

SELECT name, MAX(create\_time) AS max\_create\_time

FROM PhotocopyPoint.TypeService

GROUP BY name

) AS latest\_services

ON TS.name = latest\_services.name AND TS.create\_time = latest\_services.max\_create\_time

INNER JOIN PhotocopyPoint.Price PP ON TS.price\_id = PP.id

INNER JOIN PhotocopyPoint.Money PM ON PP.money\_id = PM.id

WHERE TS.name LIKE CONCAT('%', “д”, '%')

ORDER BY TS.id DESC;

Отримати тип послуги та його ціну за назвою яка починається на «д»

SELECT TS.id,

TS.name,

TS.info,

TS.term,

PM.count,

PM.unit,

TS.create\_time

From PhotocopyPoint.Money PM

inner join PhotocopyPoint.Price PP on PM.id = PP.money\_id

inner join PhotocopyPoint.TypeService TS on PP.id = TS.price\_id

where TS.name like CONCAT('%', “д”, '%')

ORDER BY TS.create\_time DESC LIMIT 1;

Отримати усі статуси створення замовлення.

SELECT PT.id,

PT.order\_id,

GROUP\_CONCAT(TS.name SEPARATOR ', ') AS all\_names,

PT.isUsed,

Ci.full\_name,

Ci.phone\_number

FROM PhotocopyPoint.Ticket PT

INNER JOIN PhotocopyPoint.`Order` PO ON PO.id = PT.order\_id

INNER JOIN PhotocopyPoint.Client C ON PO.client\_id = C.id

INNER JOIN PhotocopyPoint.Contact\_info Ci ON C.contact\_info\_id = Ci.id

INNER JOIN PhotocopyPoint.TypeServiceOrder TSO ON PO.id = TSO.order\_id

INNER JOIN PhotocopyPoint.TypeService TS ON TSO.type\_service\_id = TS.id

GROUP BY PT.id, PT.order\_id, PT.isUsed, Ci.full\_name, Ci.phone\_number

ORDER BY PT.id ASC;

Отримання ціни типу з назвою «Копіювання»:

SELECT PhotocopyPoint.Money.count, PhotocopyPoint.Money.unit, PhotocopyPoint.Price.create\_time

From PhotocopyPoint.Money

inner join PhotocopyPoint.Price on PhotocopyPoint.Price.money\_id = PhotocopyPoint.Money.id

inner join PhotocopyPoint.TypeService on PhotocopyPoint.TypeService.price\_id = PhotocopyPoint.Price.id

where PhotocopyPoint.TypeService.name = "Копіювання"

ORDER BY PhotocopyPoint.Price.create\_time DESC LIMIT 1;

Отримання оператора принтера на станції за адресою «Київ, вул. Хрещатик, 1», який зараз не зайнятий:

SELECT PhotocopyPoint.Contact\_info.id, PhotocopyPoint.Contact\_info.full\_name

From PhotocopyPoint.Contact\_info

inner join PhotocopyPoint.Staff on PhotocopyPoint.Staff.contact\_info\_id = PhotocopyPoint.Contact\_info.id

inner join PhotocopyPoint.PickUpStationStaff on PhotocopyPoint.PickUpStationStaff.staff\_id = PhotocopyPoint.Staff.id

left join PhotocopyPoint.OrderStaff on PhotocopyPoint.OrderStaff.staff\_id = ALL(Select PhotocopyPoint.Staff.id from PhotocopyPoint.Staff inner join PhotocopyPoint.PickUpStation on PhotocopyPoint.PickUpStation.id = PhotocopyPoint.PickUpStationStaff.pickUpStation\_id WHERE PhotocopyPoint.PickUpStation.address = "missingValue" )

left join PhotocopyPoint.`Order` on PhotocopyPoint.`Order`.id = PhotocopyPoint.OrderStaff.order\_id

left join PhotocopyPoint.TypeServiceOrder on PhotocopyPoint.TypeServiceOrder.id = PhotocopyPoint.`Order`.`id`

where (PhotocopyPoint.TypeServiceOrder.term < NOW() or (PhotocopyPoint.TypeServiceOrder.term is null)) LIMIT 1;

Отримати час створення замовлення під назвою «Копіювання»:

SELECT PhotocopyPoint.TypeService.term, PhotocopyPoint.TypeService.create\_time

From PhotocopyPoint.TypeService

where TypeService.name = "Копіювання"

ORDER BY PhotocopyPoint.TypeService.create\_time DESC LIMIT 1;

Отримання додаткової інформації щодо типу замовлення під назвою «Копіювання»:

SELECT PhotocopyPoint.TypeService.name, PhotocopyPoint.TypeService.info, PhotocopyPoint.TypeService.create\_time From PhotocopyPoint.TypeService

where PhotocopyPoint.TypeService.name = "Копіювання"

ORDER BY PhotocopyPoint.TypeService.create\_time DESC LIMIT 1;

Отримати усіх клієнтів, які мають по батькові Іванов:

Select Contact\_info.full\_name, Contact\_info.phone\_number, `Client`.station\_name

from `Client`

inner join Contact\_info on `Client`.contact\_info\_id = Contact\_info.id

where Contact\_info.full\_name like '%Іванов%';

Отримати замовлення, у який термін закінчення створення в заданий період:

SELECT \* FROM `TypeServiceOrder` where TypeServiceOrder.term BETWEEN '2024-05-27 08:00:0' and '2024-05-27 18:00:00';

Скільки створено замовлень за ім’ям даного клієнта:

SELECT count(\*) FROM `Order` inner join Client on Client.id = `Order`.id INNER JOIN Contact\_info on Client.contact\_info\_id = Contact\_info.id WHERE Contact\_info.full\_name LIKE '%Іван%';

Скільки кожний робітник виконав замовлень за весь час:

SELECT Contact\_info.full\_name, COUNT(OrderStaff.id) From Contact\_info

inner join Staff on Staff.contact\_info\_id = Contact\_info.id

inner join PickUpStationStaff on PickUpStationStaff.staff\_id = Staff.id

left join OrderStaff on Staff.id = OrderStaff.staff\_id

left join `Order` on `Order`.id = OrderStaff.order\_id

left join TypeServiceOrder on TypeServiceOrder.id = `Order`.`id`

where TypeServiceOrder.term < NOW() or (TypeServiceOrder.term is null)

GROUP BY Contact\_info.full\_name ORDER BY Contact\_info.full\_name;

Отримати терміни виготовлення замовлення для типу замовлення «Друк»:

SELECT TypeServiceOrder.term from TypeServiceOrder where TypeServiceOrder.type\_service\_id = ALL(SELECT TypeService.`id` from TypeService WHERE TypeService.name = 'Друк');

Отримати всі ім’я та, якщо є, суму потрачену на замовлення:

SELECT Contact\_info.full\_name, (

SELECT SUM(Money.count)

FROM Money

inner join `Order` on `Order`.`money\_id` = Money.id

inner join `Client` on `Order`.client\_id = `Client`.id

WHERE `Client`.`contact\_info\_id` = Contact\_info.id

) AS total\_money

FROM Contact\_info;

Хто з робітників не повинен виконувати замовлення у цьому місяці:

SELECT Contact\_info.full\_name   
FROM Contact\_info  
INNER JOIN Staff on Staff.contact\_info\_id = Contact\_info.id  
  
LEFT JOIN   
(select staff\_id  from OrderStaff    
inner JOIN `Order` on `Order`.`id` = OrderStaff.order\_id  
inner JOIN TypeServiceOrder on TypeServiceOrder.order\_id = `Order`.`id`  
where (TypeServiceOrder.term  BETWEEN DATE\_FORMAT(NOW(), "%y-%m-01") and LAST\_DAY(NOW()))) as ss    
on ss.staff\_id = Staff.id  
where ss.staff\_id  is null;

Отримати усі ціни типи послуг за деякою додатковою інформацією окрім перерахованих:

SELECT TypeService.name, Money.`count`, Money.unit from Money

INNER JOIN Price on Price.money\_id = Money.id

INNER JOIN TypeService on TypeService.price\_id = Price.id

WHERE TypeService.info not in ("Чорно-білий друк")

GROUP BY TypeService.name, Money.`count`, Money.unit ORDER BY TypeService.name;

Хто з робітників ні разу не отримував завдання робити замовлення:

SELECT Contact\_info.full\_name FROM Contact\_info

INNER JOIN Staff on Staff.contact\_info\_id = Contact\_info.id

where not EXISTS (SELECT OrderStaff.order\_id FROM OrderStaff

WHERE OrderStaff.staff\_id = Staff.id);

Хто з робітників отримав максимальну кількість замовлення і хто не отримав жодного за весь час:

SELECT name, MAX(count\_staff), ‘Хто з робітників отримав максимальну кількість замовлення за весь час’ FROM (SELECT Contact\_info.full\_name as name, COUNT(OrderStaff.id) as count\_staff From Contact\_info

inner join Staff on Staff.contact\_info\_id = Contact\_info.id

inner join PickUpStationStaff on PickUpStationStaff.staff\_id = Staff.id

left join OrderStaff on Staff.id = OrderStaff.staff\_id

left join `Order` on `Order`.id = OrderStaff.order\_id

left join TypeServiceOrder on TypeServiceOrder.id = `Order`.`id`

where TypeServiceOrder.term < NOW()

GROUP BY Contact\_info.full\_name ORDER BY Contact\_info.full\_name) as t

GROUP BY name

UNION

SELECT name, count\_staff, ‘хто не отримав жодного за весь час’

from (SELECT Contact\_info.full\_name as name, 0 as count\_staff FROM Contact\_info

INNER JOIN Staff on Staff.contact\_info\_id = Contact\_info.id

where not EXISTS (SELECT OrderStaff.order\_id FROM OrderStaff

WHERE OrderStaff.staff\_id = Staff.id)) as l

GROUP BY name ORDER BY name;