

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з комп'ютерного практикуму № 2 з дисципліни
«Бази даних»

„Створення бази даних”

Виконала

IT-01 Дмитрієва Ірина
(шифр, прізвище, ім'я, по батькові)

Перевірила

Марченко О.І.
(прізвище, ім'я, по батькові)

Київ 2021

Мета:

- Створення бази даних шляхом визначення схеми БД та заповнення її тестовими даними
- Навчитися проектувати бази даних, вводити і редагувати структуру таблиць та дані в таблицях
- Вивчити команди SQL для роботи з таблицями (створення, зміни та видалення таблиць)
- Вивчити використовувані в SQL засоби для підтримки цілісності даних та їх практичне застосування

Теоретичні основи:

Викладені в лекційному матеріалі

Постановка задачі комп'ютерного практикуму № 2

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Створити схему БД згідно з розробленою в роботі №1 ER-моделлю
- 2) Розробити SQL-скрипти для:

а. створення таблиць в БД засобами мови SQL (для 5-6 таблиць).

Передбачити необхідність наявності обмежень для підтримки цілісності даних;

б. зміни структури таблиць та даних в таблицях засобами мови SQL (до 20 різних за суттю запитів для 5-6 таблиць);

с. видалення окремих елементів або самих таблиць засобами мови SQL (до 10 різних за суттю команд);

д. встановлення зв'язків між таблицями засобами мови SQL.

- 3) Згенерувати схему даних засобами СУБД

- 4) Імпортувати дані в створену БД з використанням засобів СУБД

Варіант індивідуального завдання – 7:

Програмне забезпечення готелю

Основна задача програмного забезпечення, котре проєктується - є відстеження фінансової сторони роботи готелю. Діяльність організована наступним чином: готель надає номери клієнтам на певний термін. Кожен номер характеризується місткістю, комфортністю (люкс, напівлюкс, звичайний , тощо) і ціною. Клієнтами є різні особи,, по яким збирається певна інформація (прізвище, ім'я, по-батькові, номер та серія паспорту або іншого документу, що посвідчує особу, додаткова інформація). Здача номера клієнтові проводиться за наявності вільних місць в номерах, за параметрами, котрі вказав клієнт. При поселенні фіксується дата поселення, при виїзді - дата звільнення. номеру. Необхідно не лише зберігати інформацію за фактом здачі номера клієнтові, але і здійснювати бронювання номерів. Крім того, для постійних клієнтів, а також для певних категорій клієнтів передбачена система знижок. Знижки можуть підсумовуватися.

SQL-скрипти створення таблиць у MySQL:

1) category_type

Сутність Категорії пільгової знижки category_type дозволяє додавати знижки для особливих груп Клієнтів (наприклад, студент, пенсіонер, військовослужбовець). Знижки будуть підсумовуватися в залежності від описаної накопичувальної системи (постійного користувача) і пільгової.

Заздалегідь передбачено, що тип знижки та сума не повинні дорівнювати null.

```
CREATE TABLE IF NOT EXISTS hotel.category_type (  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    `type` INT NOT NULL,  
    discount FLOAT NOT NULL);
```

2) client_status

Категорія постійного користувача client_status є сутністю, яка містить назву категорії (наприклад, Початковий, Середній, Преміум, Бізнес), діапазон витрачених коштів, що й визначають дану категорію, атрибут про знижку.

Передбачено, що всі атрибути не можуть бути null.

```
CREATE TABLE IF NOT EXISTS hotel.client_status (  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    spent_money_from INT NOT NULL,  
    spent_money_to INT NOT NULL,  
    discount FLOAT NOT NULL,  
    `name` VARCHAR(45) NOT NULL  
);
```

3) client

Клієнт містить інформацію про прізвище, ім'я, по-батькові, тип, номер та серію документу, що посвідчує особу, додаткову інформацію. Також при створенні таблиці були встановлені зовнішні ключі з таблиць категорії постійного користувача client_status та категорії пільгової знижки category_type через їхні id, щоб клієнт мав інформацію про свій статус.

Передбачено, що атрибути імені та прізвища не можуть бути null. Тип та серія документу повинні бути обов'язково вказані, серія по дефолту дорівнює нулю.

```
CREATE TABLE IF NOT EXISTS hotel.client (  
    id int PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    first_name varchar(45) NOT NULL,  
    second_name varchar(45) DEFAULT '-',  
    last_name varchar(45) NOT NULL,
```

```
document_type          enum('Паспорт', 'ID-карта', 'Закордонний  
паспорт', 'Посвідчення водія') NOT NULL,  
document_number int NOT NULL,  
document_series varchar(5) DEFAULT NULL,  
details varchar(45) NOT NULL);
```

4) room

Сутність Готельний номер містить атрибути про місткість, комфортність (подано через Enum: люкс, напівлюкс, звичайний, економ) і дефолтна ціна (без впливу додаткових знижок), номер кімнати.

```
CREATE TABLE IF NOT EXISTS hotel.room (  
  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  `number` VARCHAR(45) NOT NULL,  
  capacity DECIMAL(1,0) NOT NULL,  
  comfort VARCHAR(10) NOT NULL,  
  default_price FLOAT NOT NULL);
```

5) booking

Сутність Бронювання містить інформацію про Клієнта, Готельний номер, дати бронювання: чек-ін та чек-аут, реальний час поселення та виселення з готелю, статус бронювання (створено, активно, завершено, скасовано). Таким чином за допомогою наявних бронювань можливо визначати список вільних готельних номерів за бажаними датами та параметрами. Сутність Сплати посилається на Бронювання. Припускається, що по одному бронюванню може бути кілька сплат (депозит під час створення бронювання, решта при заселенні, можливі додаткові сервіси).

```
CREATE TABLE IF NOT EXISTS hotel.booking (  
  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  `check-in_date` DATE NOT NULL,  
  `check-out_date` DATE NOT NULL,  
  room_id INT NOT NULL DEFAULT NULL,  
  `status` ENUM  
  ('Скасовано', 'Активно', 'Завершено') NOT NULL,  
  arrival_date DATETIME DEFAULT '0000-00-00 00:00:00',  
  departure_date DATETIME DEFAULT '0000-00-00 00:00:00');
```

6) payments

Сутність Сплати містить в атрибутах номер кредитної картки, дату оплати, інформацію про надану знижку та повну актуальну ціну на заданий період бронювання, враховуючи специфічні знижки або підвищення цін. Ця інформація необхідна як реальний сплачений рахунок, адже загальна ціна за

номер готелю може змінюватися з часом. Також при створенні таблиці був встановлений зовнішній ключ з таблиці бронювання

```
CREATE TABLE IF NOT EXISTS hotel.payments (  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    room_charge FLOAT NOT NULL,  
    credit_number INT NULL DEFAULT NULL,  
    discount_amount FLOAT NOT NULL,  
    payment_date DATE NOT NULL);
```

З сутностей Бронювання та Сплати можна розрахувати загальну кількість витрачених коштів по кожному Клієнту за весь час по всіх його бронюванням. Це важливо для оновлення його Категорії постійного користувача (повинне виконуватися після кожної сплати).

SQL-скрипти для зміни структури таблиць та даних в таблицях у MySQL:

1) Зміна таблиці категорії постійного користувача: встановлення дефолтних значень замість null.

```
ALTER TABLE hotel.client_status  
MODIFY spent_money_from INT DEFAULT (0),  
MODIFY spent_money_to INT DEFAULT (0),  
MODIFY discount INT DEFAULT (0);
```

2) Зміна таблиці клієнта, встановлення дефолтного значення серії документа.

```
ALTER TABLE hotel.client  
MODIFY document_series VARCHAR(5) DEFAULT '-';
```

3) Зміна таблиці готельного номеру: додавання нової колонки.

```
ALTER TABLE hotel.room  
ADD details VARCHAR(12);
```

4) Зміна таблиці готельного номеру: збільшення кількості можливих уведених символів.

```
ALTER TABLE hotel.room  
MODIFY details VARCHAR(200);
```

5) Додавання додаткової таблиці знижки за днем тижня.

```
CREATE TABLE IF NOT EXISTS hotel.discount_per_day(  

```

```
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT);
```

6) Зміна таблиці знижки за днем тижня: додавання нових полів.

```
ALTER TABLE hotel.discount_per_day  
ADD `day` VARCHAR(10),  
ADD discount INT;
```

7) Зміна таблиці знижки за днем тижня: обмеження допустимих введених значень – NOT NULL.

```
ALTER TABLE hotel.discount_per_day  
MODIFY `day` VARCHAR(10) NOT NULL;
```

8) Зміна значень таблиці готельного номеру.

```
SELECT * FROM hotel.room;
```

	id	number	capacity	comfort	default_price
►	1	301	3	Common	50
	2	350	2	Half lux	100
	3	333	3	Lux	200
	4	100	1	Half lux	150
	5	106	2	Common	50
	6	261	4	Common	50
	7	600	2	Half lux	150
	8	500	3	Half lux	150
	9	111	1	Lux	200
	10	122	2	Common	50
	11	200	3	Half lux	150
	12	330	3	Half lux	100

```
UPDATE hotel.room  
SET comfort = 'Lux', default_price = '200'  
WHERE id = 1;
```

	id	number	capacity	comfort	default_price
►	1	301	3	Lux	200
	2	350	2	Half lux	100
	3	333	3	Lux	200
	4	100	1	Half lux	150
	5	106	2	Common	50
	6	261	4	Common	50
	7	600	2	Half lux	150
	8	500	3	Half lux	150
	9	111	1	Lux	200
	10	122	2	Common	50
	11	200	3	Half lux	150
	12	330	3	Half lux	100

9) Зміна значень таблиці готельного номеру.

```
SELECT * FROM hotel.room;
```

	id	number	capacity	comfort	default_price
►	1	301	3	Lux	200
	2	350	2	Half lux	100
	3	333	3	Lux	200
	4	100	1	Half lux	150
	5	106	2	Common	50
	6	261	4	Common	50
	7	600	2	Half lux	150
	8	500	3	Half lux	150
	9	111	1	Lux	200
	10	122	2	Common	50
	11	200	3	Half lux	150
	12	330	3	Half lux	100

```
UPDATE hotel.room  
SET default_price = '250'  
WHERE comfort = 'Lux';
```

	id	number	capacity	comfort	default_price
►	1	301	3	Lux	250
	2	350	2	Half lux	100
	3	333	3	Lux	250
	4	100	1	Half lux	150
	5	106	2	Common	50
	6	261	4	Common	50
	7	600	2	Half lux	150
	8	500	3	Half lux	150
	9	111	1	Lux	250
	10	122	2	Common	50
	11	200	3	Half lux	150
	12	330	3	Half lux	100

10) Зміна значень таблиці бронювання.

```
UPDATE hotel.booking  
SET departure_date = NULL  
WHERE `status` = 'Active';
```

```
UPDATE hotel.booking  
SET departure_date = NULL,  
arrival_date = NULL, room_id = NULL  
WHERE `status` = 'Canceled';
```


SQL-скрипти для видалення окремих елементів або самих таблиць засобами мови SQL (до 10 різних за суттю команд):

1) Видалення необов'язкового поля у таблиці готельного номеру:

```
ALTER TABLE hotel.room  
DROP details;
```

2) Видалення зовнішнього ключа таблиці знижки за день.

```
ALTER TABLE hotel.payments  
DROP FOREIGN KEY payments_discountperday;
```

3) Видалення полів таблиці знижки за день.

```
ALTER TABLE hotel.discount_per_day  
DROP COLUMN day,  
DROP COLUMN discount;
```

4) Видалення таблиці знижки за день.

```
DROP TABLE hotel.discount_per_day;
```

5) Видалення певних значень з таблиці готельного номеру.

```
SELECT * FROM hotel.room WHERE id >= 20;
```

	id	number	capacity	comfort	default_price
►	20	255	2	Common	50
	21	269	2	Common	50
	22	329	4	Common	50
	23	229	4	Common	50
	24	222	1	Lux	200
	25	307	4	Common	50
	26	340	2	Half lux	100
	27	321	2	Common	50
	28	444	2	Lux	200
	29	400	3	Half lux	150
	30	300	3	Half lux	150
*	NULL	NULL	NULL	NULL	NULL

```
DELETE FROM hotel.room  
WHERE id > 25;
```

	id	number	capacity	comfort	default_price
▶	20	255	2	Common	50
	21	269	2	Common	50
	22	329	4	Common	50
	23	229	4	Common	50
	24	222	1	Lux	200
	25	307	4	Common	50
*	NULL	NULL	NULL	NULL	NULL

SQL-скрипти встановлення зв'язків у MySQL:

1) Додавання поля у таблицю клієнта для створення зовнішнього ключа з таблиці категорії постійного користувача.

```
ALTER TABLE hotel.client
ADD category_type_id INT NOT NULL;

----

ALTER TABLE hotel.client
ADD CONSTRAINT client_categorytype
FOREIGN KEY (category_type_id)
REFERENCES hotel.category_type (id)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

2) Додавання поля у таблицю клієнта для створення зовнішнього ключа з таблиці статусу клієнта.

```
ALTER TABLE hotel.client
ADD client_status_id INT NOT NULL;

----

ALTER TABLE hotel.client
ADD CONSTRAINT client_clientstatus
FOREIGN KEY (client_status_id)
REFERENCES hotel.client_status (id)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

3) Додавання поля у таблицю сплат для створення зовнішнього ключа з таблиці бронювання.

```
ALTER TABLE hotel.payments
ADD booking_id INT NULL;

----
```

```
ALTER TABLE hotel.payments
ADD CONSTRAINT payments_booking
    FOREIGN KEY (booking_id)
    REFERENCES hotel.booking (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

4) Додавання поля у таблицю бронювання для створення зовнішнього ключа з таблиці готельного номеру. Набуває значення NULL, якщо статус Created.

```
ALTER TABLE hotel.booking
ADD room_id INT NULL;
```

```
ALTER TABLE hotel.booking
ADD CONSTRAINT booking_room
    FOREIGN KEY (room_id)
    REFERENCES hotel.room (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

5) Додавання поля у таблицю бронювання для створення зовнішнього ключа з таблиці клієнта.

```
ALTER TABLE hotel.booking
ADD client_id INT NOT NULL;
```

```
ALTER TABLE hotel.booking
ADD CONSTRAINT booking_client
    FOREIGN KEY (client_id)
    REFERENCES hotel.`client` (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

6) Додавання поля у таблицю сплат для створення зовнішнього ключа з тимчасової таблиці знижки за днем тижня.

```
ALTER TABLE hotel.payments
ADD discount_per_day_id INT NOT NULL;
```

```
ALTER TABLE hotel.payments
ADD CONSTRAINT payments_discountperday
    FOREIGN KEY (discount_per_day_id)
    REFERENCES hotel.discount_per_day (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

Імпортування даних у створену БД з використанням засобів СУБД:

Додавання рядка зі значеннями у таблицю категорії постійного користувача.

```
INSERT INTO hotel.category_type (type, discount)
VALUES ('standart', '0');
```

Заповнення таблиці готельного номеру даними із файлу:

```
LOAD DATA INFILE
'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/room.csv'
INTO TABLE hotel.room
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

SELECT * FROM hotel.room;
```

Заповнення таблиці категорії постійного користувача даними із файлу:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/category_type.csv'
INTO TABLE hotel.category_type
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

SELECT * FROM hotel.category_type;
```

Заповнення таблиці статусу клієнта даними із файлу:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/client_status.csv'
INTO TABLE hotel.client_status
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

SELECT * FROM hotel.client_status;
```

Заповнення таблиці клієнта даними із файлу:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/client.csv'
INTO TABLE hotel.client
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'

```

```

    LINES TERMINATED BY '\r\n'
    IGNORE 1 ROWS;

```

```

SELECT * FROM hotel.client;

```

Заповнення таблиці бронювання даними із файлу:

```

    LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/booking.csv'
    INTO TABLE hotel.booking
    CHARACTER SET utf8mb4
    FIELDS TERMINATED BY ';'
    LINES TERMINATED BY '\r\n'
    IGNORE 1 ROWS;

```

```

SELECT * FROM hotel.booking;

```

Заповнення таблиці сплат даними із файлу:

```

    LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/payments.csv'
    INTO TABLE hotel.payments
    CHARACTER SET utf8mb4
    FIELDS TERMINATED BY ';'
    LINES TERMINATED BY '\r\n'
    IGNORE 1 ROWS;

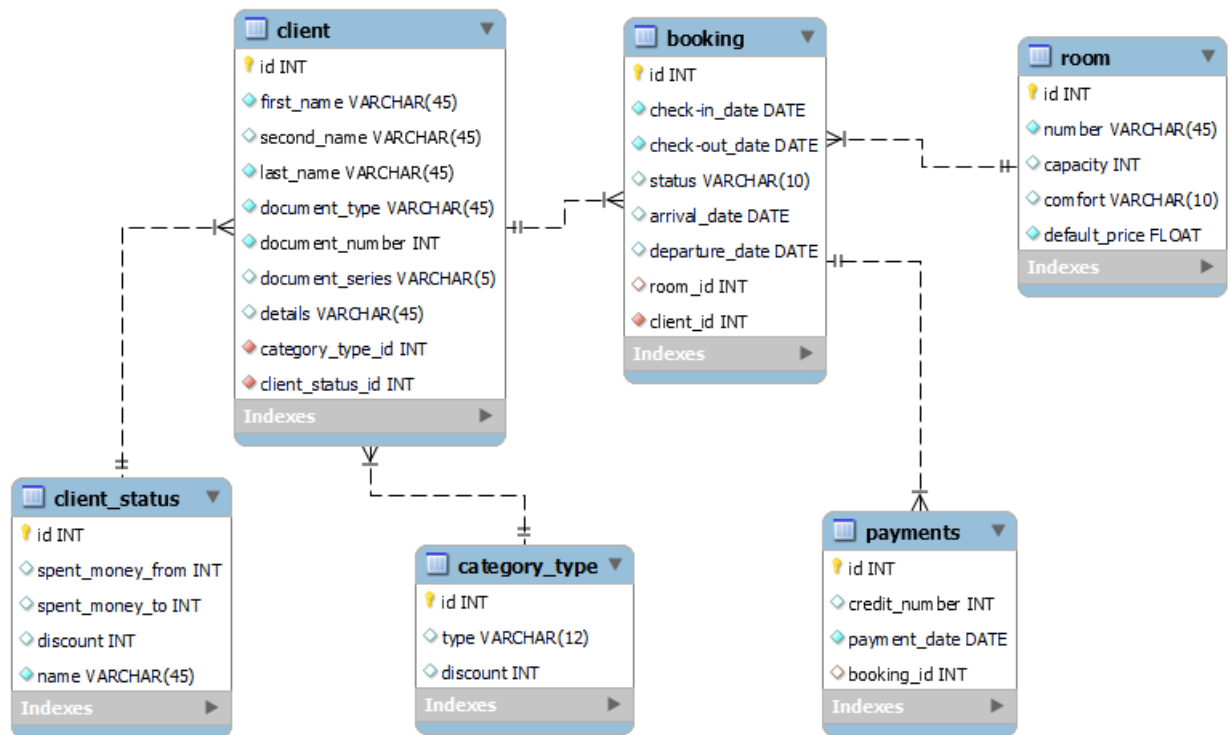
```

```

SELECT * FROM hotel.payments;

```

Схема спроектованої бази даних



Висновок:

у даній лабораторній роботі було створено базу даних шляхом визначення схеми БД та заповнення її тестовими даними; було набуто навичок проектування бази даних, введення й редагування структури таблиць та даних в таблицях; було вивчено команди SQL для роботи з таблицями (створення, зміни та видалення таблиць) SQL засоби, що використовуються для підтримки цілісності даних, використано на практиці.