КУРСОВА РОБОТА

база даних сервісного центру МВС в частині прийому екзаменів, водійських посвідчень

Виконав студент групи
ІП-22 Нижник Дмитро

МЕТА РОБОТИ

Мета даної курсової роботи – реалізація бази даних для підтримки роботи сервісного центру МВС в частині прийому екзаменів, водійських посвідчень.

База даних повинна забезпечувати ефективну роботу сервісного центру при прийомі практичних та теоретичних екзаменів, отриманні та оновленні водійських посвідчень, має полегшувати менеджмент транспорту сервісного центру.

ПОСТАНОВКА ЗАВДАННЯ

01. АНАЛІЗ
предметного середовища, існуючих продуктів та бізнес–правил

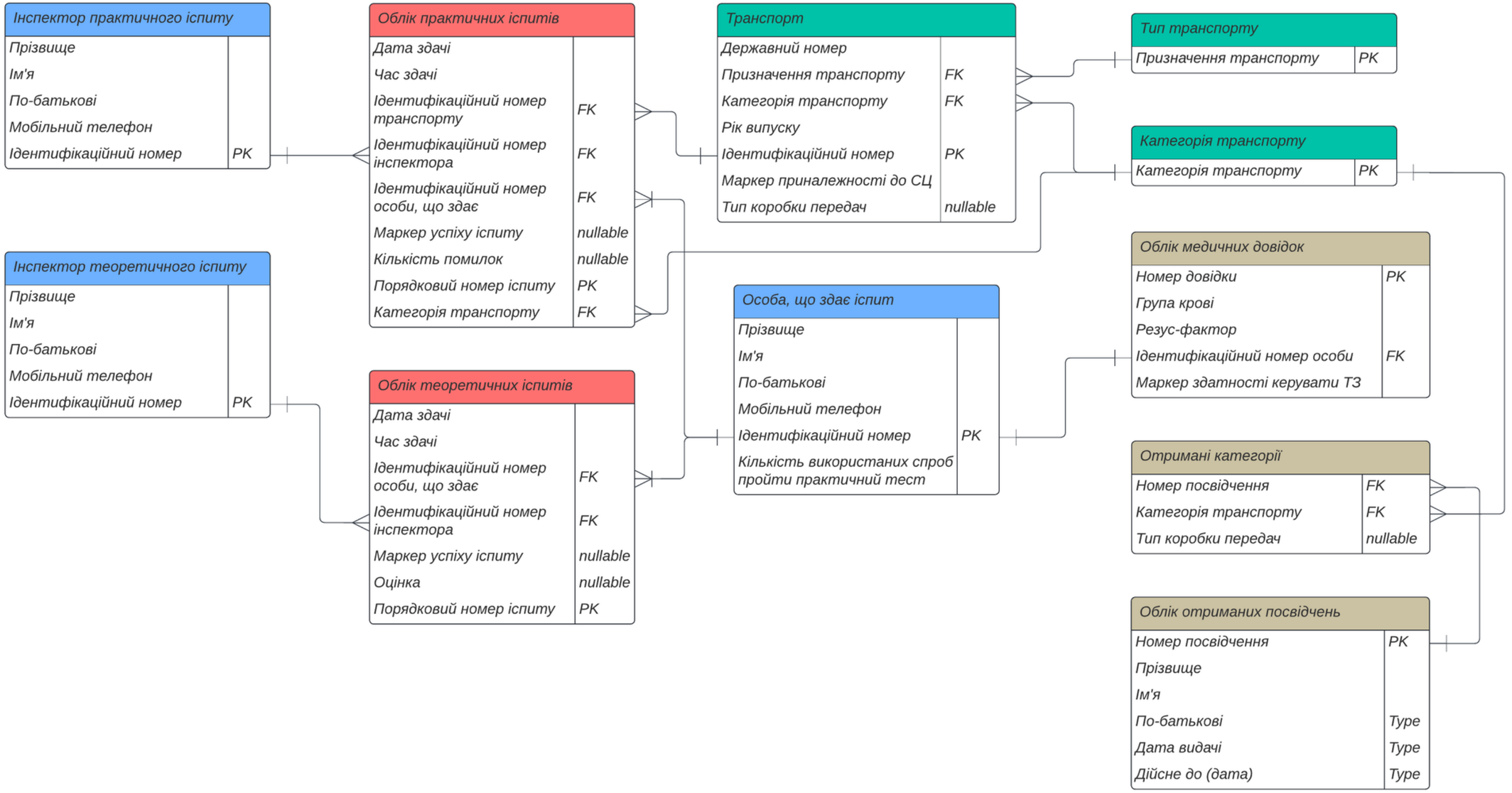
02. РОЗРОБКА
діаграми “сутність–зв’язок” та реляційної моделі

03. РЕАЛІЗАЦІЯ
бази даних, тригерів, запитів, функцій та процедур, представлень

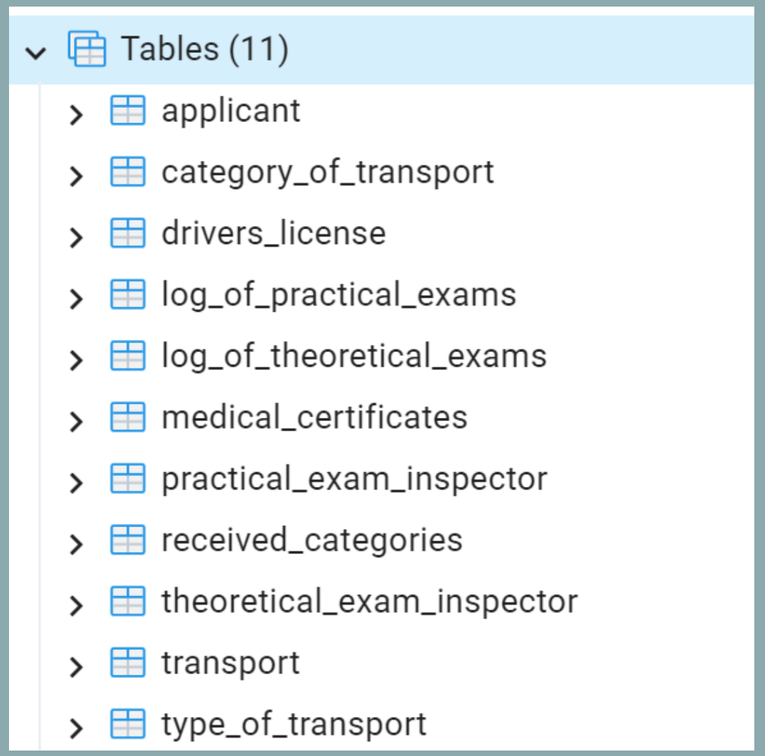
04. ОПТИМІЗАЦІЯ
роботи бази даних

05. ВИСНОВКИ
аналіз проведеної роботи

ERD


Інспектор практичного іспиту
Прізвище
Ім'я
По-батькові
Мобільний телефон
Ідентифікаційний номер
PK
Облік практичних іспитів
Дата здачі
Час здачі
Ідентифікаційний номер транспорту
FK
Ідентифікаційний номер інспектора
FK
Ідентифікаційний номер особи, що здає
FK
Маркер успіху іспиту
nullable
Кількість помилок
nullable
Порядковий номер іспиту
PK
Категорія транспорту
FK
Транспорт
Державний номер
Призначення транспорту
FK
Категорія транспорту
FK
Рік випуску
Ідентифікаційний номер
PK
Маркер приналежності до СЦ
Тип коробки передач
nullable
Тип транспорту
Призначення транспорту
PK
Категорія транспорту
Категорія транспорту
PK
Інспектор теоретичного іспиту
Прізвище
Ім'я
По-батькові
Мобільний телефон
Ідентифікаційний номер
PK
Облік теоретичних іспитів
Дата здачі
Час здачі
Ідентифікаційний номер особи, що здає
FK
Ідентифікаційний номер інспектора
FK
Маркер успіху іспиту
nullable
Оцінка
nullable
Порядковий номер іспиту
PK
Особа, що здає іспит
Прізвище
Ім'я
По-батькові
Мобільний телефон
Ідентифікаційний номер
PK
Кількість використаних спроб пройти практичний тест
Облік медичних довідок
Номер довідки
PK
Група крові
Резус-фактор
Ідентифікаційний номер особи
FK
Маркер здатності керувати ТЗ
Отримані категорії
Номер посвідчення
FK
Категорія транспорту
FK
Тип коробки передач
nullable
Облік отриманих посвідчень
Номер посвідчення
PK
Прізвище
Ім'я
По-батькові
Type
Дата видачі
Type
Дійсне до (дата)
Type

РЕАЛІЗАЦІЯ БАЗИ ДАНИХ




```
service_center_mia=# \dt
                  List of relations
 Schema |              Name              | Type  |  Owner
--------+--------------------------------+-------+----------
 public | applicant                      | table | postgres
 public | category_of_transport          | table | postgres
 public | drivers_license                | table | postgres
 public | log_of_practical_exams         | table | postgres
 public | log_of_theoretical_exams       | table | postgres
 public | medical_certificates           | table | postgres
 public | practical_exam_inspector       | table | postgres
 public | received_categories            | table | postgres
 public | theoretical_exam_inspector     | table | postgres
 public | transport                      | table | postgres
 public | type_of_transport              | table | postgres
(11 rows)
```

```
postgres=# CREATE DATABASE service_center_mia;
CREATE DATABASE
postgres=# \l
                                                    List of databases
       Name        |  Owner   | Encoding | Locale Provider |          Collate          |           Ctype           | ICU Locale | ICU Rules |   Access privileges
-------------------+----------+----------+-----------------+---------------------------+---------------------------+------------+-----------+-----------------------
 postgres          | postgres | UTF8     | libc            | English_United States.1251 | English_United States.1251 |            |           |
 service_center_mia | postgres | UTF8     | libc            | English_United States.1251 | English_United States.1251 |            |           |
 template0         | postgres | UTF8     | libc            | English_United States.1251 | English_United States.1251 |            |           | =c/postgres          +
                   |          |          |                 |                           |                           |            |           | postgres=CTc/postgres
 template1         | postgres | UTF8     | libc            | English_United States.1251 | English_United States.1251 |            |           | =c/postgres          +
                   |          |          |                 |                           |                           |            |           | postgres=CTc/postgres
(4 rows)
```

РОЛІ ТА КОРИСТУВАЧІ

Для роботи з базою даних було створено 4 ролі та 4 користувачі на основі цих ролей.

Створені ролі:
1) Головний менеджер сервісного центру
2) Менеджер документів та екзаменів
3) Іспектор
4) Особа, що здає іспит

```
-- Створення ролей --
CREATE ROLE document_exam_manager;
CREATE ROLE chief_manager;
CREATE ROLE applicant;
CREATE ROLE inspector;
```

```
-- Створення користувачів --
CREATE USER chief_manager_user WITH PASSWORD 'admin';
GRANT chief_manager TO chief_manager_user;
CREATE USER document_exam_manager_user WITH PASSWORD 'password';
GRANT document_exam_manager TO document_exam_manager_user;
CREATE USER inspector_user WITH PASSWORD '244466666';
GRANT inspector TO inspector_user;
CREATE USER applicant_user WITH PASSWORD '987654321';
GRANT applicant TO applicant_user;
```

РОЛІ ТА КОРИСТУВАЧІ

```
service_center_mia=# SELECT rolname FROM pg_roles;
        rolname
----------------------
 postgres
 document_exam_manager
 chief_manager
 applicant
 inspector
(5 rows)
```

```
service_center_mia=# SELECT usename FROM pg_user;
          usename
---------------------------
 postgres
 chief_manager_user
 document_exam_manager_user
 inspector_user
 applicant_user
(5 rows)
```

```
service_center_mia=# \c service_center_mia applicant_user
Password for user applicant_user:
You are now connected to database "service_center_mia" as user "applicant_user".
service_center_mia=> CALL insert_theoretical_exam('2024-01-06', '17:00', 'ZY099QM7M6', 'EF56');
ERROR:  permission denied for table log_of_theoretical_exams
```

```
service_center_mia=> \c service_center_mia inspector_user
Password for user inspector_user:
You are now connected to database "service_center_mia" as user "inspector_user".
service_center_mia=> CALL update_ability_to_drive('1JLNREGYNG', FALSE);
ERROR:  permission denied for table medical_certificates
```

РЕАЛІЗАЦІЯ ТРИГЕРІВ

```sql
-- Функція для тригера на вставку елемента в drivers_license --
CREATE OR REPLACE FUNCTION check_practical_exam_success()
RETURNS TRIGGER AS $$
BEGIN
    IF NOT EXISTS (
        SELECT 1
        FROM log_of_practical_exams pe
        JOIN applicant a ON pe.applicant_id = a.applicant_id
        WHERE a.last_name = NEW.last_name
          AND a.first_name = NEW.first_name
          AND a.middle_name = NEW.middle_name
          AND pe.success_marker = TRUE
    )
    THEN RAISE EXCEPTION 'Applicant has not passed the practical exam! The insertion was aborted!';
    ELSE RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;
-- 4 Тригер, що спрацьовує при вставці того кандидата, який не склав теоретичний іспит
CREATE OR REPLACE TRIGGER check_practical_exam_trigger
BEFORE INSERT ON drivers_license
FOR EACH ROW
EXECUTE FUNCTION check_practical_exam_success();
```

```
service_center_mia=# INSERT INTO drivers_license (drivers_license_number, last_name, first_name, middle_name, obtaining_date, expiration_date)
service_center_mia-# VALUES ('RTY128716', 'Hrytsenko', 'Daria', 'Mykolaivna', '2023-12-30', '2053-12-30');
ERROR:  Applicant has not passed the practical exam! The insertion was aborted!
```

ЗБЕРЕЖУВАНІ ФУНКЦІЇ ТА ПРОЦЕДУРИ

```sql
-- 8 Вставка тоеретичного іспиту--
CREATE OR REPLACE PROCEDURE insert_theoretical_exam(
    p_date DATE,
    p_time TIME,
    p_inspector_id VARCHAR(4),
    p_applicant_id VARCHAR(10))
AS $$ DECLARE v_category_of_transport VARCHAR(2);
BEGIN
    IF NOT is_theoretical_inspector_available(p_date, p_time) THEN
        RAISE EXCEPTION 'Too many appointments at this time! Choose another.';
    ELSE INSERT INTO log_of_theoretical_exams (date_of_passing, time_of_passing, applicant_id, inspector_id, success_marker, exam_grade)
         VALUES (p_date, p_time, p_inspector_id, p_applicant_id, NULL, NULL);
         RAISE NOTICE 'Inserted!';
    END IF;
END;
$$ LANGUAGE plpgsql;
```

```
service_center_mia=# CALL insert_theoretical_exam('2024-01-06', '17:00', '2REUWXLKQF', 'EF56');
ERROR:  Applicant cannot drive a vehicle or does not have a medical certificate! The insertion was aborted!
CONTEXT:  PL/pgSQL function check_ability_to_drive() line 8 at RAISE
SQL statement "INSERT INTO log_of_theoretical_exams (date_of_passing, time_of_passing, applicant_id, inspector_id, success_marker, exam_grade)
        VALUES (p_date, p_time, p_inspector_id, p_applicant_id, NULL, NULL)"
PL/pgSQL function insert_theoretical_exam(date,time without time zone,character varying,character varying) line 5 at SQL statement
service_center_mia=# CALL insert_theoretical_exam('2024-01-06', '17:00', 'ZY099QM7M6', 'EF56');
NOTICE:  Inserted!
CALL
```

ПРЕДСТАВЛЕННЯ ТА ЗАПИТИ

```sql
-- 2 Представлення, яке містить у собі теоретичні іспити, що заплановані на сьогоднішній день --
CREATE OR REPLACE VIEW theoretical_exams_today AS
SELECT te.exam_serial_number,
       te.time_of_passing,
       a.last_name || ' ' || a.first_name || ' ' || a.middle_name AS full_name,
       te.applicant_id,
       te.inspector_id,
       te.success_marker,
       te.exam_grade
FROM log_of_theoretical_exams te
JOIN applicant a ON te.applicant_id = a.applicant_id
WHERE te.date_of_passing = CURRENT_DATE
ORDER BY te.exam_serial_number;
-- Відкриття та видалення представлення
SELECT * FROM theoretical_exams_today;
DROP VIEW IF EXISTS theoretical_exams_today;
```

```sql
-- 4 Виводить ті теоретичні екзамени, на які не з'явилися кандидати --
SELECT lte.date_of_passing, lte.time_of_passing, a.last_name, a.first_name, a.middle_name
FROM log_of_theoretical_exams lte
JOIN applicant a ON lte.applicant_id = a.applicant_id
WHERE lte.date_of_passing < CURRENT_DATE
  AND lte.success_marker IS NULL;
```

ОПТИМІЗАЦІЯ

```
-- Створення індексів --
CREATE INDEX idx_practical_exam_inspector_id ON practical_exam_inspector(inspector_id);
CREATE INDEX idx_theoretical_exam_inspector_id ON theoretical_exam_inspector(inspector_id);
CREATE INDEX idx_applicant_id ON applicant(applicant_id);
CREATE INDEX idx_transport_id ON transport(transport_id);
CREATE INDEX idx_log_practical_exam_serial_number ON log_of_practical_exams(exam_serial_number);
CREATE INDEX idx_log_theoretical_exam_serial_number ON log_of_theoretical_exams(exam_serial_number);
CREATE INDEX idx_medical_certificate_number ON medical_certificates(certificate_number);
CREATE INDEX idx_drivers_license_number ON drivers_license(drivers_license_number);
```

Таблиця 12.1 – Аналіз впливу індексів на першому запиті

Номер спроби	1	2	3	4	5
Час виконання до додання індексу, мс	0.147	0.136	0.135	0.125	0.126
Час виконання після додання індексу, мс	0.117	0.121	0.120	0.114	0.110

ВИСНОВКИ

ПРОВЕДЕНО

– аналіз предметної області
– аналіз бізнес–процесів
– аналіз вже існуючих продуктів

РОЗРОБЛЕНО

– ER діаграму
– реляційну модель
– базу даних засобами PostgreSQL

СТВОРЕНО

– ролі та користувачів
– збережувані функції та процедури
– запити та представлення
– тригери

ОКРІМ ТОГО

– базу даних оптимізовано
– підведено підсумки роботи
– створено документацію

ДЯКУЮ ЗА УВАГУ!