**Презентація: База даних для сервісного центру МВС**

**Вступ**

Добрий день! Я, Віжуткін Ілля, студент групи ІП-31. Темою моєї курсової роботи — була розробка бази даних для підтримки роботи сервісного центру МВС в частині прийому іспитів та видачі посвідчень. Це є досить актуальним завданням, адже автоматизація в таких центрах може значно підвищити якість та швидкість обслуговування громадян. Сьогодні я розповім, як була побудована ця база даних і як вона вирішує основні задачі сервісного центру.

**Аналіз предметного середовища**

Починалось все із аналізу предметного середовища, під час якого було визначено, що одну із важливих функцій, яку виконують сервісні центри МВС – це прийом екзаменів та видача водійських посвідчень. Тут клієнти проходять через кілька основних етапів, таких як: реєстрація, складання теоретичного та практичного іспитів та отримання посвідчень.

Проблема полягає в тому, що ці процеси потребують значних ресурсів для обробки інформації, такої як, внесення клієнтів, перевірка їх та їх документів, обробка записів на іспити, саме іспитування, видача посвідчень та їх контроль, а також контроль окремих сервісних центрів, теоретичних матеріалів, інспекторів та автомобілів.

Автоматизація цих операцій дозволяє значно полегшити роботу працівників центру та забезпечити точність і швидкість обслуговування клієнтів.

Для кращого розуміння необхідних компонентів бази даних, також було проаналізовано такі системи, як DMV у США та WORD у Польщі. Цей аналіз допоміг визначити функціональні особливості, які можна використати або вдосконалити у розробленій базі даних.

**Постановка завдання**

Отже, метою цього проєкту було створення бази даних, яка підтримуватиме ключові процеси сервісного центру МВС у частині прийому іспитів та видачі посвідчень. У межах роботи було реалізовано такі основні етапи:

* Спочатку необхідно було описати основні **бізнес-правила** для майбутьної системи.
* Потім треба було проаналізувати необхідні сутності, їх взаємозв’язки та на основі цього побудвати **ER-діаграму**.
* Далі, на основі існуючого опису та діаграми, необхідно було реалізувати **справжню базу даних**.
* І під кінець, необхідно було провести певну роботу з базою даних, а саме: реалізувати представлення, функції, процедури, тригери, select-запити та провести оптимізацію

**Основні бізнес-правила**

Під час першого етапу було визначено та реалізовано такі основні бізнес-правила:

* Обмеження по віку клієнта
* Необхідність документів
* Перевірка, що клієнт отримує нову категорію
* Обмеження на один активний запис
* Обмеження в 10 днів між записами
* Перевірка, що за одним талоном створено тільки один іспит
* Перевірка наявності необхідного автомобіля
* Перевірка, що обраний автомобіль відповідає інформації, зазначеній в талоні
* Перевірка відповідності інспектора
* Автоматична зміна статусів талонів
* Видача посвідчення після успішного складання іспитів
* Перевірка дати та часу будь-яких записів на адекватність

**ER-модель**

Під час другого етапу, проаналізувавши предметне середовище та аналоги, було побудовано ER-модель, що включає основні необхідні сутності та зв'язки між ними. В цілому, ця модель, що можна побачити на фото, забезпечує логічну, досить компактну, чітку та ефективну структуру даних.

**Вибір СУБД**

Далі, для третього етапу, а саме реалізації бази даних, спочатку було необхідно обрати СУБД. В моїй роботі було обрано PostgreSQL. Цей вибір був зумовлений такими її перевагами як:

* Що PostgreSQL є повністю безкоштовною.
* Разом з цим, СУБД забезпечує високий рівень безпеки через гнучку систему ролей і прав.
* Також PostgreSQL підтримує роботу зі складними запитами та типами даних.
* Також важливо, що ця система може обробляти дуже великі обсяги даних.
* Ну і не менш важливо, що PostgreSQL має одну з найбільших спільнот серед СУБД з відкритим кодом.

**Реалізація бази даних**

Отже, база даних була реалізована в СУБД PostgreSQL. Таблиці було створено у відповідності до ER-діаграми. Цілісність даних підтримується за допомогою первинних і зовнішніх ключів, а також обмежень UNIQUE та CHECK. Для реалізації складніших бізнес-правил, як вже було сказано, пізніше були створені тригери. Крім того, було створено відповідних користувачів з різними рівнями доступу до різних частин системи.

**Робота з базою даних**

На останньому етапі, для автоматизації бази даних та спрощення взаємодії з нею, було розроблено такий функціонал:

* **А саме, представлення**, що забезпечують зручний доступ до агрегованої або фільтрованої інформації.
* **Функції** **та процедури** для автоматизації повторюваних дій та обробки результатів із різними вхідними даними.
* **Тригери**, що автоматизують виконання складних бізнес-правил.
* **Та Select-запити,** що надають необхідну інформацію для функціонування сервісних центрів.

**Приклади представлень**

Далі неведу кілька прикладів. Ось, наприклад, деякі із створених представлень.

* Представлення all\_tickets об’єднує дані про все теоретичні та практичні талони клієнтів.
* А представлення theory\_exam\_result\_info, в свою чергу, надає детальну інформацію про теоретичні іспити, клієнтів, питання та відповіді.

**Приклади функцій та процедур**

Далі подивимось на деякі приклади функцій та процедур.

* Наприклад, функція get\_client\_age очислює вік клієнта, використовуючи його дату народження.
* Далі бачимо приклад представлення update\_expired\_tickets\_status, що оновлює статуси талонів, які не були використані протягом вказаного дня.
* Ну і інша, функція get\_new\_theory\_exam\_questions. Ця функція генерує новий набір випадкових питань для теоретичного іспиту клієнта, уникаючи тих, на які він вже відповідав.

**Приклади тригерів**

Далі йдуть приклади тригерів

* Наприклад, тригер check\_client\_has\_nessesery documents перевіряє, чи клієнт має необхідні документи для запису на теоретичний чи практичний іспит.
* Іншим прикладом є тригер check\_client\_age, що аналогічно перевіряє вік клієнта перед записом на будь-який іспит.

**Приклади Select-запитів**

На останок, подивимось на деякі приклади Select-запитів.

* Першим таким є запит для підрахунку спроб теоретичного іспиту для кожного клієнта. Цей запит підраховує невдалі спроби, які набрали менше 18 балів.
* Другим прикладом бачимо запит, що аналізує популярність типів автомобілей, а саме кількість практичних іспитів для кожної категорії та коробки передач транспортних засобів.

**Оптимізація**

І так, останнім, що було зроблено в даній роботі була оптимізація за допомогою індексів. Оскільки, основні індекси на первинні та зовнішні ключі та унікальні поля були створені автоматично за допомогою СУБД, то нам залишилось створити індекси тільки на деякі додаткові поля, що можуть бути використані при роботі сервісних центрів.

Провівши тестування на комплексі запитів, було виявлено, що індексація прискорила швидкість приблизно в 5 разів навіть на невеликих об’ємах даних. На більших об’ємах, очевидно, що індекси покращать швидкодію значно більше.

**Висновок**

Отже, розроблена база даних для сервісного центру МВС автоматизує ключові процеси пов’язані із іспитуванням та видачою посвідчень, така система підвищує ефективність обслуговування та зменшує ризик людьских помилок. Також вона може бути масштабована та інтегрована з іншими державними системами в майбутньому.

Дякую за увагу!