

КУРСОВА РОБОТА
КР.ІПЗ-16.00.000 ПЗ
Група ІПЗ-2018
Слободян Юрій

2020

Івано-Франківський університет
Короля Данила

Кафедра програмної інженерії та інформаційних технологій

КУРСОВА РОБОТА

з дисципліни ОРГАНІЗАЦІЯ БАЗ ДАНИХ

Тема – Розробка демонстраційного програмного забезпечення
Оператори та конструкції мови програмування Assembler

ПОЯСНУВАЛЬНА ЗАПИСКА

КР.ІІЗ – 00.00.000. ІІЗ

Студент гр.ІІЗ-2018 _____ (Слободян Ю. В.)
(підпис)

Допускається до захисту
Керівник курсового проекту

(посада) (підпис) (дата) (розшифровка підпису)

м. Івано-Франківськ
2020

Івано-Франківський університет Короля Данила

Кафедра програмної інженерії та інформаційних технологій

Дисципліна – Організація баз даних

Спеціальність – Інженерія програмного забезпечення

Курс - 3 Група - ІПЗ-2018 Семестр - 5

ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ

Студенту Слободяну Юрію Віталійовичу

1. Тема проекту - Розробка демонстраційного програмного забезпечення, оператори та конструкції мови програмування `Assembler`.
2. Термін здачі студентом закінченого проекту – 14 грудня 2020 року.
3. Зміст пояснювальної записки:
 - опис основних інформаційних характеристик даних необхідних для організації бази даних операторів та конструкцій мови програмування `Assembler`;
 - розробка структури бази даних, опис структури та функціонального призначення полів таблиць бази даних, а також формування зв'язків між таблицями;
 - розробка демонстраційного програмного забезпечення на основі мови `SQL`, що забезпечує виконання операцій доповнення, редагування, видалення та пошуку даних в таблицях розробленої бази.
4. Дата видачі завдання - 15 вересня 2020 року

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів курсової роботи	Термін виконання етапів роботи	Примітка
1. Встановлення локального сервера та налаштування середовища розробки.	25.09.2020	
2. Створення бази даних і таблиць до неї.	30.09.2020	
3. Наповнення таблиць контентом відповідно до заданої теми курсової роботи. Формування зв'язків між таблицями.	09.10.2020	
4. Створення папочної структури курсової роботи в середовищі розробки. Під'єднання бази даних до веб-сторінки.	14.10.2020	
5. Верстання основних компонентів курсової роботи на веб-сторінці (таблиці, заголовки таблиць, компоненти вводу даних та кнопки підтвердження).	26.10.2020	
6. Динамічне заповнення таблиць веб-сторінки даними з таблиць, які створені в базі даних.	02.11.2020	
7. Реалізація додавання нового запису в таблиці бази даних через веб-сторінку. Тестування, виправлення помилок.	11.11.2020	
8. Реалізація редагування записів таблиць бази даних через веб-сторінку. Тестування, виправлення помилок.	17.11.2020	
9. Реалізація видалення записів таблиць бази даних через веб-сторінку. Тестування, виправлення помилок.	21.11.2020	
9. Загальна стилізація веб-сторінки, перевірка роботи програмного забезпечення на помилки.	04.12.2020	
10. Опис курсової роботи та заповнення календарного плану.	05.12.2020	

Студент

(підпис)

(розшифровка підпису)

Керівник проекту

(підпис)

(розшифровка підпису)

Анотація

У курсовому проекті було розроблено базу даних, та допоміжне програмне забезпечення для відображення інформації про основні оператори та конструкції мови програмування Assembler. Для розробки бази даних було використано реляційну базу даних MySQL. Допоміжне програмне забезпечення розроблене за допомогою веб-технологій HTML5, CSS3, PHP.

У першому розділі описано вимоги до проектування та характерні особливості інформаційного процесу та способи представлення даних, які беруться за основу при проектуванні структури бази даних.

У другому розділі здійснений опис розробки структури бази даних, пояснення щодо інформаційного призначення кожної таблиці. Подано структуру бази даних.

У третьому розділі завдяки мові запитів SQL розроблене демонстраційне програмне забезпечення, що дозволяє виконувати типові операції роботи з базою даних.

У четвертому розділі продемонстровано роботу демонстраційного програмного забезпечення.

Summary

The course project developed a database and auxiliary software for displaying information about the main operators and constructions of the Assembler programming language. A MySQL relational database was used to develop the database. Utility software developed using web technologies HTML5, CSS3, PHP.

The first section describes the design requirements and characteristics of the information process and methods of data presentation, which are taken as a basis for designing the database structure.

The second section describes the development of the database structure, an explanation of the information purpose of each table. The structure of the database is given.

In the third section, thanks to the SQL query language, demo software has been developed that allows you to perform typical database operations.

The fourth section demonstrates the operation of demonstration software.

ЗМІСТ

ВСТУП.....	7 ст
1. Вимоги до проектування бази даних.....	9 ст
2. Розробка структури бази даних.....	12 ст
2.1 Проектування бази даних.....	12 ст
2.2 Створення таблиць.....	14 ст
3. Опис розробки програми	16 ст
3.1 Огляд виводу даних	16 ст
3.2 Розробка демонстраційної програми	17 ст
4 Опис роботи програми	21 ст
ВИСНОВКИ	23 ст
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	24 ст

ВСТУП

База даних – це набір даних з наступними властивостями:

- дані логічно пов'язані між собою і несуть відповідну інформацію;
- структура баз даних звичайно відповідає тому специфічному набору даних, які вона містить;

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

Системи управління даними першого покоління

СУБД першого покоління характерні тим, що кожна група користувачів розробляла своє власне програмне забезпечення по управлінню даними. Наслідками такої сепаратизації стало надмірне дублювання програмних кодів і даних.

Системи управління даними другого покоління

Файли взаємопов'язаних даних об'єднуються в бази даних. СУБД створюються для таких досвідчених користувачів, як програмісти.

Системи управління даними третього покоління

Переваги СУБД :

- Скорочення надлишку даних;
- Без баз даних неможливо уникнути зберігання надлишкових даних;
- При наявності центрального контролю баз даних деякі надлишкові дані можна усунути;
- Надлишкові дані не можуть бути повністю усунені, оскільки велику роль в СУБД відіграють питання часу і достовірності.

У світі існує безліч СУБД. Незважаючи на те, що вони можуть по-різному працювати з різними об'єктами і надають користувачу різні функції й засоби, більшість СУБД спираються на єдиний устояний комплекс основних понять. Це дає нам можливість розглянути одну систему й узагальнити її поняття, прийоми й методи на весь клас СУБД. В якості такого навчального об'єкта розглянемо СУБД Microsoft Access, що входить до пакета Microsoft Office.

Найпоширенішими СУБД є Visual FoxPro, PostgreSQL, MySQL та Microsoft Access.

1 ВИМОГИ ДО ПРОЕКТУВАННЯ БАЗИ ДАНИХ

Головні ідеї сучасних інформаційних технологій базуються на концепції баз даних. Відповідно до цієї концепції, основою інформаційних технологій є дані, які повинні бути організовані в бази даних в цілях адекватного відображення мінливого реального світу і задоволення інформаційних потреб користувачів.

Одним з найважливіших і основоположних понять в теорії баз даних є поняття інформації. Під інформацією розуміються будь-які відомості про будь-яку подію, процеси, об'єкти. У свою чергу, дані - це інформація, представлена в певному вигляді, що дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом. Для комп'ютерних технологій дані - це інформація в дискретному, фіксованому вигляді, зручна для зберігання, обробки на ЕОМ, а також для передачі по каналах зв'язку.

Для коректного визначення вимог, необхідних для проектування бази даних операторів та конструкцій мови програмування Assembler, визначимо такі поняття. База даних (БД) - це сукупність взаємозв'язаних даних при такій мінімальній надмірності, яка допускає їх використання оптимальним чином для одного або декількох додатків в певній предметній області.

Система управління базами даних (СУБД) - сукупність мовних та програмних засобів, призначених для створення, ведення і сумісного використання БД багатьма користувачами.

У базі даних інформація повинна бути організована так, щоб забезпечити мінімальну частку її надмірності, оскільки надмірність даних тягне за собою ряд негативних наслідків: збільшення обсягу інформації, а отже, потреба в додаткових ресурсах для зберігання і обробки, а так само поява помилок при введенні дублюючої інформації, що порушують цілісність бази даних.

В БД повинні зберігатися дані, логічно пов'язані між собою. Для того щоб дані можна було пов'язати між собою, і зв'язати так, щоб ці зв'язки відповідали

реально існуючим в даній предметній області, останню піддають детальному аналізу, виділяючи сутності чи об'єкти. Сутність або об'єкт - це те, про що необхідно зберігати інформацію. Сутності мають деякі характеристики, звані атрибутами, які теж необхідно зберігати в БД.

Визначивши сутності і їхні атрибути, необхідно перейти до виявлення зв'язків, які можуть існувати між деякими сутностями. Зв'язок - це те, що об'єднує дві або більше сутностей. Зв'язки між сутностями також є частиною даних, і вони також повинні зберігатися в базі даних.

Проектована база даних повинна володіти певними властивостями.

- Цілісність бази даних досягається внаслідок введення обмежень, пов'язані з нормалізацією БД. У кожен момент часу існування БД відомості, що містяться в ній, повинні бути несуперечливі.
- Відновлюваність - можливість відновлення БД після збою системи або окремих видів полумок системи.
- Безпека - припускає захист даних від навмисного і ненавмисного доступу, модифікації або руйнування.
- Ефективність. Властивість ефективності зазвичай оцінюється двома параметрами: мінімальний час реакції на запит користувача і мінімальні потреби в пам'яті, а так же поєднанням цих параметрів.

Для програмної реалізації бази даних були обрані мови SQL і php, і відповідно, програмне забезпечення.

Розглянута не процедурна мова SQL (Structured Query Language - структурована мова запитів) орієнтована на операції з даними, представленими у вигляді логічно взаємопов'язаних сукупностей таблиць.

Особливість пропозицій цієї мови полягає в тому, що вони орієнтовані більшою мірою на кінцевий результат обробки даних, ніж на процедуру цієї обробки. SQL сам визначає, де знаходяться дані, які індекси і навіть які найбільш ефективні послідовності операцій слід використовувати для їх отримання.

Згідно індивідуальним завданням, мета даної роботи полягає у проектуванні та розробці бази даних для відображення інформації про оператори та конструкції мови програмування Assembler.

Для реалізації поставленої мети ми скористаємося двома методами для проектування БД. Першим підходом до проектування є метод сутність - зв'язок, другим же є метод нормалізації відносин.

Модель сутність - зв'язок або по іншому ER-модель - це модель даних, що дозволяє описувати концептуальні схеми предметної області. Дана модель використовується при високорівневому проектуванні баз даних і дозволяє з її допомогою виділити ключові сутності і позначити зв'язки, які встановлюються між цими сутностями.

Нормалізація відносин - це формальний апарат обмежень на формування відносин, який дозволяє усунути дублювання і потенційну суперечливість збережених даних, таким чином зменшує трудовитрати на ведення БД. Процес нормалізації полягає в декомпозиції вихідних відносин на більш прості відносини, до тих пір, поки не вийде такий проект БД, в якому «кожен факт з'являється лише в одному місці».

2 РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ

2.1 Проектування бази даних

На першому етапі проектування необхідно визначити сутності. Сутність - це якийсь об'єкт, що представляє інтерес для користувача.

З точки зору заданої предметної області типовим користувачем розроблюваної бази даних буде людина, яка хоче увійти в курс основ мови програмування Assembler. Основне завдання даної бази даних саме забезпечити всією необхідною інформацією. Користувачем даної БД може бути людина, яка починає вивчати дану мову програмування з нуля, так як в базі буде міститися інформація про оператори та конструкції мови програмування Assembler.

			id_constr	uncondi_transitions	condi_transitions	loops_construct				
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	JMP	JN	LOOPZ
<input type="checkbox"/>		Изменить		Копировать		Удалить	2		JE	CYCL
<input type="checkbox"/>		Изменить		Копировать		Удалить	3		JNE	LOOP
<input type="checkbox"/>		Изменить		Копировать		Удалить	4		JL	LOOPNE
<input type="checkbox"/>		Изменить		Копировать		Удалить	5		JG	LOOPE
<input type="checkbox"/>		Изменить		Копировать		Удалить	6		JGE	LOOPNZ
<input type="checkbox"/>		Изменить		Копировать		Удалить	7		JB	
<input type="checkbox"/>		Изменить		Копировать		Удалить	8		JBE	
<input type="checkbox"/>		Изменить		Копировать		Удалить	9		JA	
<input type="checkbox"/>		Изменить		Копировать		Удалить	10		JAE	

			id_oper	logical_oper	arithmetic_oper	data_transferring_oper				
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	AND	ADD	MOV
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	OR	ADC	XCHG
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	XOR	SUB	PUSH
<input type="checkbox"/>		Изменить		Копировать		Удалить	4	TEST	MUL	POP
<input type="checkbox"/>		Изменить		Копировать		Удалить	5	NOT	IMUL	LEA
<input type="checkbox"/>		Изменить		Копировать		Удалить	6		DIV	
<input type="checkbox"/>		Изменить		Копировать		Удалить	7		IDIV	
<input type="checkbox"/>		Изменить		Копировать		Удалить	8		DEC	
<input type="checkbox"/>		Изменить		Копировать		Удалить	9		INC	
<input type="checkbox"/>		Изменить		Копировать		Удалить	10		NEG	

	id_meaning	arithmetic_oper_meaning	logical_oper_meaning	data_transferring_oper_meaning	uncondi_transition_meaning	condi_transition_meaning	loops_constr_meaning
1	Arithmetic operations - perform operations of addi...	Logical operations - perform logical actions on op...	Common forward operators are commands that forward...	In the constructions of unconditional transition, ...	Conditional jump command - analyzes the values o...	A loop is a repetitive execution c a sequence of ...	
2	mov ax, [P:P]C-PSPSP [®] ; P:P] P:P]CЪC-PIP [®] C;CЪCЪ...	MOV AL, 10110101b; XOR AL, 0; MOV AL, 10110...	mov ax bx;	code segment jmp short go ; go: code ends	CMP R7, R8; JNE TONI	metka: int 21h; inc di; loc metka; ...	

Рисунок 2.1 – Загальна структура бази даних

Для досягнення мети і вирішення поставленого завдання виділимо на основі аналізу предметної області наступні сутності: `assembler_constructions`(Конструкції Асемблера), `assembler_operators`(Оператори Асемблера), `meaning_oper_constr`(Визначення та приклади конструкцій та операторів).

Другий етап - визначення зв'язків між виділеними сутностями. `meaning_oper_constr`(Визначення та приклади конструкцій та операторів) містить посилання на `assembler_constructions`(Конструкції Асемблера) та посилання на `assembler_operators`(Оператори Асемблера).

Далі визначимо атрибути або властивості для кожної з сутностей, а так само ключі для кожної сутності. Ключем сутності називається атрибут або набір атрибутів, що дозволяють однозначно ідентифікувати примірник сутності:

`assembler_constructions`(Конструкції Асемблера):

- `Id_constr`(Ідентифікатор) – внутрішній ключ
- `uncondi_transitions`(безумовний перехід) – типу `text(50)`
- `condi_transitions`(перехід з умовою) – типу `text(50)`
- `loops_construct`(конструкції циклів) – типу `text(50)`

`assembler_operators`(Оператори Асемблера):

- `id_oper`(Ідентифікатор) – внутрішній ключ
- `logical_oper`(Логічні оператори) – типу `text(50)`
- `arithmeticl_oper`(Арифметичні оператори) – типу `text(50)`
- `data_transferring_oper`(Оператори пересилання даних) – типу `text(50)`

meaning_oper_constr(Визначення та приклади конструкцій і операторів):

- id_meaning(Ідентифікатор) - внутрішній ключ
- arithmetic_oper_meaning(Визначення арифметичних операцій) - типу text(50)
- logical_oper_meaning(Визначення логічних операцій) - типу text(50)
- data_transferring_oper_meaning(Визначення операцій пересилання даних) - типу text(50)
- uncondi_transition_meaning(Визначення безумовного переходу) - типу text(50)
- condi_transition_meaning(Визначення переходу з умовою) - типу text(50)
- loops_constr_meaning(Визначення конструкцій циклу) - типу text(50)

2.2 Створення таблиць

Таблиці складають основу бази даних - саме в них зберігаються всі дані. Перед усім, повинна бути спланована структура кожної таблиці. Структура таблиць обумовлюється вмістом тих вихідних форм, запитів та звітів, які повинні бути отримані при роботі з базою даних.

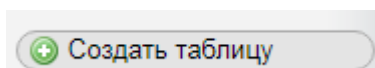
При плануванні таблиць необхідно уникати повторення колонок в різних таблицях, тільки якщо вони не слугують для визначення зв'язків між ними.

Таблиця складається з записів (рядків), кожний з яких описує одну сутність. Кожна колонка таблиці - це поле. Поле містить однотипну інформацію, яка визначає тип даних.

Тип даних визначає вид і межі допустимих значень, які можуть бути введені в поле, а також об'єм пам'яті, який виділяється для цього поля, що важливо при проектуванні великих БД.

В створеній базі даних, створюємо таблиці за допомогою phpMyAdmin.

Створення таблиці `assembler_constructions`



Структура

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	Δ_I	Комментарии	MIME-тип	Преобразование	Параметры преобразований
	INT		Нет				---					
	INT		Нет				---					
	INT		Нет				---					
	INT		Нет				---					

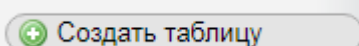
Комментарий к таблице:

Тип таблиц: Сравнение:

Определение разделов (PARTITION):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 <code>id_constr</code>	int(11)	Нет	Нет			AUTO_INCREMENT	
<input type="checkbox"/>	2 <code>uncondi_transitions</code>	text	utf8_general_ci	Нет	Нет			
<input type="checkbox"/>	3 <code>condi_transitions</code>	text	utf8_general_ci	Нет	Нет			
<input type="checkbox"/>	4 <code>loops_construct</code>	text	utf8_general_ci	Нет	Нет			

Створення таблиці assembler_operators

 Создать таблицу

Структура

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	A_I	Комментарии	MIME-тип	Преобразование	Параметры преобразований
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				

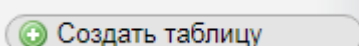
Комментарий к таблице:

Тип таблиц: Сравнение:

Определение разделов (PARTITION):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id_oper	int(5)	Нет	Нет	Нет	Нет	AUTO_INCREMENT	
<input type="checkbox"/>	2 logical_oper	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	3 arithmetic_oper	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	4 data_transferring_oper	text	utf8_general_ci	Нет	Нет	Нет		

Створення таблиці meaning_oper_constr

 Создать таблицу

Структура

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	A_I	Комментарии	MIME-тип	Преобразование	Параметры преобразований
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>				

Комментарий к таблице:

Тип таблиц: Сравнение:

Определение разделов (PARTITION):

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	1 id_meaning	int(11)	Нет	Нет	Нет	Нет	AUTO_INCREMENT	
<input type="checkbox"/>	2 arithmetic_oper_meaning	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	3 logical_oper_meaning	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	4 data_transferring_oper_meaning	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	5 uncondi_transition_meaning	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	6 condi_transition_meaning	text	utf8_general_ci	Нет	Нет	Нет		
<input type="checkbox"/>	7 loops_constr_meaning	text	utf8_general_ci	Нет	Нет	Нет		

3 ОПИС РОЗРОБКИ ПРОГРАМИ

3.1 Огляд виводу даних

У процесі проектування бази даних був розроблений ряд запитів, що забезпечує додавання, оновлення та видалення інформації, що зберігається в базі даних. Ці запити надають можливість в зручній формі переглянути інформацію про оператори та конструкції мови програмування Assembler.

Запит для виводу всієї інформації з таблиці “assembler_constructions”, результат якого можна переглянути на Рисунку 3.1

```
SELECT * FROM `assembler_constructions` ORDER BY `assembler_constructions`  
.`uncondi_transitions` DESC LIMIT 0, 30;
```

ID	Безумовний перехід	Перехід з умовою	Конструкції циклів
1	JMP	JN	LOOPZ
2		JE	CYCL
3		JNE	LOOP
4		JL	LOOPNE
5		JG	LOOPE
6		JGE	LOOPNZ
7		JB	
8		JBE	
9		JA	
10		JAЕ	

Рисунок 3.1 – Вивід даних таблиці “assembler_constructions”

Запит для виводу всієї інформації з таблиці “assembler_operators”, результат якого можна переглянути на Рисунку 3.2

```
SELECT * FROM `assembler_operators` ORDER BY `assembler_operators`.  
`id_oper` ASC LIMIT 0, 30;
```

ID	Логічні оператори	Арифметичні оператори	Оператори пересилання даних
1	AND	ADD	MOV
2	OR	ADC	XCHG
3	XOR	SUB	PUSH
4	TEST	MUL	POP
5	NOT	IMUL	LEA
6		DIV	
7		IDIV	
8		DEC	
9		INC	
10		NEG	

Рисунок 3.2 – Вивід даних таблиці “assembler_operators”

Запит для виводу всієї інформації з таблиці “ meaning_oper_constr”, результат якого можна переглянути на Рисунку 3.3

```
SELECT * FROM `meaning_oper_constr` LIMIT 0 , 30"
```

ID	Визначення арифметичних операторів	Визначення логічних операторів	Визначення операторів пересилання даних	Визначення безумовного переходу	Визначення переходу з умовою	Визначення конструкцій циклу
1	Arithmetic operations - perform operations of addition, subtraction, multiplication, division, increase by one (increase), decrease by one (decrease), etc. These commands require one or two input operands. One output operand is formed by commands.	Logical operations - perform logical actions on operands, for example, logical AND, logical OR, operation "Inequality", clearing, inversion, various shifts (right, left, arithmetic shift, cyclic shift). These commands, like arithmetic, require one or two input operands, and they form one output operand.	Common forward operators are commands that forward operands: register - register, register - memory, memory - register, register - direct operand (number), memory - direct operand (number). General purpose data operators can be used with many operands, although there are some exceptions.	In the constructions of unconditional transition, the address of the new command can be formed in two ways. In the first method, the command itself indicates an offset relative to the current command. In the first case, the transition is called short (SHORT), remote from the current address, in the second case - close (NEAR), located in the same segment with the current address.	Conditional jump command - analyzes the values ??of the required flags, and if they are set, goes to the specified address.	A loop is a repetitive execution of a sequence of commands. The LOOP design is intended for the organization of a cycle. In this construction, one operand is the name of the label to which the transition is made. The CX register is used as a cycle counter.
2	mov ax, {змінна, де зберігається 1 число} add ax, {змінна, де зберігається 2 число} mov {змінна, де буде результат}, ax	MOV AL, 10110101b; XOR AL, 0; MOV AL, 10110111b; XOR AL, 0;	mov ax bx;	code segment jmp short go ; go: code ends	CMP R7, R8; JNE TONI	metka: int 21h; inc dl; loop metka; mov ah,09h; mov dx,press; int 21h

Рисунок 3.3 – Вивід даних таблиці “meaning_oper_constr”

3.2 Розробка демонстраційної програми

Для створення демонстраційної програми було використано мову програмування php і Базу даних MySQL.

Візуальну частину було розроблено за допомогою мови гіпер-текстової розмітки HTML5 та каскадних таблиць стилів CSS3.

Першим кроком було створення бази даних (див. додаток 1). Після чого створення таблиць і заповнення її даними(див. додаток 2).

Дальше, створював папочну структуру проекту, запитами бази даних під'єднював саму базу даних до веб-сторінки.

Підключення БД до проекту можна зробити за допомогою php-функції `mysqli_connect()`:

```
$connect = mysqli_connect('localhost', 'course_work', 'safety22DB', 'course_work');
```

Наступним кроком буде створення функціоналу для загрузки даних в базу даних(див. додаток 3). Дальше додаємо кнопку оновлення бази даних для того, щоб після її заповнення або внесення змін ці зміни збереглись в базу даних (див. додаток 4).

Тепер додаємо функціонал (`insert`, `delete`, `update`) за допомогою запитів SQL та функцій php `GET` і `POST`. Спершу з url-адреси вибираємо `id` поля таблиці. Після цього вставляємо запит SQL та підставляємо туди `id` уже витягнуте з url-адреси. Додаємо атрибут `“value”` до елементів `“input”` і в ньому прописуємо створені змінні мовою php. (див. додаток 5).

4 ОПИС РОБОТИ ПРОГРАМИ

Демонстрацію роботи програми розглянемо на прикладі таблиці `assembler_operators`

Ось такий вигляд має таблиця `assembler_operators` зображено на рисунку 4.1

ID	Логічні оператори	Арифметичні оператори	Оператори пересилання даних		
1	AND	ADD	MOV	Редагувати	Видалити
2	OR	ADC	XCHG	Редагувати	Видалити
3	XOR	SUB	PUSH	Редагувати	Видалити
4	TEST	MUL	POP	Редагувати	Видалити
5	NOT	IMUL	LEA	Редагувати	Видалити
6		DIV		Редагувати	Видалити
7		IDIV		Редагувати	Видалити
8		DEC		Редагувати	Видалити
9		INC		Редагувати	Видалити
10		NEG		Редагувати	Видалити

Рисунок 4.1 – Дані таблиці таблиця `assembler_operators`

Перевіримо оператор видалення рядка з бази даних, візьмемо оператор під індексом 10 і видалимо його, рисунок 4.2

Таблиця "assembler_operators"

ID	Логічні оператори	Арифметичні оператори	Оператори пересилання даних		
1	AND	ADD	MOV	Редагувати	Видалити
2	OR	ADC	XCHG	Редагувати	Видалити
3	XOR	SUB	PUSH	Редагувати	Видалити
4	TEST	MUL	POP	Редагувати	Видалити
5	NOT	IMUL	LEA	Редагувати	Видалити
6		DIV		Редагувати	Видалити
7		IDIV		Редагувати	Видалити
8		DEC		Редагувати	Видалити
9		INC		Редагувати	Видалити

Рисунок 4.2 – Дані таблиці assembler_operators після Delete
 Як ми бачимо такого запису більше немає в базі. Тепер поспробуємо оновити дані по id 7, рисунок 4.3.

Таблиця "assembler_operators"

ID	Логічні оператори	Арифметичні оператори	Оператори пересилання даних		
1	AND	ADD	MOV	Редагувати	Видалити
2	OR	ADC	XCHG	Редагувати	Видалити
3	XOR	SUB	PUSH	Редагувати	Видалити
4	TEST	MUL	POP	Редагувати	Видалити
5	NOT	IMUL	LEA	Редагувати	Видалити
6		DIV		Редагувати	Видалити
7		ASD		Редагувати	Видалити
8		DEC		Редагувати	Видалити
9		INC		Редагувати	Видалити
10		NEG		Редагувати	Видалити

Рисунок 4.3 – Дані таблиці assembler_operators після Update
 Як бачимо в таблиці assembler_operators поле під індексом 7 змінило своє значення і результат зберігся в базу.

Тепер можна переглянути оператор Insert, рисунок 4.4

Таблиця "assembler_operators"

ID	Логічні оператори	Арифметичні оператори	Оператори пересилання даних		
1	AND	ADD	MOV	Редагувати	Видалити
2	OR	ADC	XCHG	Редагувати	Видалити
3	XOR	SUB	PUSH	Редагувати	Видалити
4	TEST	MUL	POP	Редагувати	Видалити
5	NOT	IMUL	LEA	Редагувати	Видалити
6		DIV		Редагувати	Видалити
7		IDIV		Редагувати	Видалити
8		DEC		Редагувати	Видалити
9		INC		Редагувати	Видалити
10		NEG		Редагувати	Видалити
11	some text1	some text2	some text3	Редагувати	Видалити

Додавання нового запису в таблицю

Логічні оператори

Арифметичні оператори

Оператори пересилання даних

Додати

Рисунок 4.4 – Дані таблиці assembler_operators після Insert

ВИСНОВКИ

Курсова робота закріпила знання з основ сучасної технології створення складних програмних продуктів з використанням баз даних. Такі знання можуть бути використані у розробках як системного, так і прикладного програмного забезпечення з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та ефективності програмних продуктів, які створюються.

Для виконання курсової роботи використались знання та практичні навички, що отримані на лекціях та лабораторних заняттях.

Використовуючи набуті знання створено базу даних, яка відображає інформацію про оператори на конструкції мови програмування *Assembler*, та демонстраційне програмне забезпечення, яке полегшує роботу з базою даних.

Курсова робота була виконана з застосуванням мови програмування *php*.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основна:

1. Дейт, К.Дж. Введение в системы баз данных, 7-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002.-1072с..
2. Смирнов С.Н., Задворев И.С. Работаем с Oracle 9i – Х.: Изд-во при Харьк. унив-те, Фолио 2003.
3. Юрчишин В.М., Клим Б.В., Кропивницька В.Б. Організація баз даних. / Навчальний посібник – Івано-Франківськ: Факел, 2009.

Додаткова:

1. <https://metanit.com/>;
2. <https://www.youtube.com/watch?v=GDVWdYZLM7w&t=2460s>
3. <http://blogs.kpi.kharkov.ua/v2/asm/wp-content/uploads/sites/20/2015/12/Sistemo-programmirovaniye-Rysovanyj-2008.pdf>
4. <https://www.sites.google.com/site/sistprogr/lekcii1/lek9>
5. <https://studfile.net/preview/3730800/page:10/>