

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Факультет інформатики та обчислювальної техніки

Кафедра комп'ютерних
систем та мереж

МЕТОДИЧНІ ВКАЗІВКИ
до самостійної роботи
з дисципліни “Організація баз даних”
для студентів спеціальності 123 “Комп'ютерна інженерія”
усіх форм навчання
частина I

2019

Методичні вказівки до самостійної роботи з дисципліни “Організація баз даних” для студентів спеціальності 123 “Комп’ютерна інженерія” усіх форм навчання частина I /Укл.: Т.О. Паромова, - Запоріжжя: НУ ”Запорізька політехніка”, 2019 – 24 с.

Укладачі: Т.О.Паромова, ст.викладач,

Рецензент: С.Д. Точилін, доцент, к.ф.-м.н.

Відповідальний за випуск: Паромова Т.О., ст.викладач.

Затверджено
на засіданні кафедри
“Комп’ютерні системи
та мережі ”
Протокол №4 от 15.11.2019

Рекомендовано до видання
НМК факультету
комп’ютерних
наук і технологій
Протокол №3 от 18.11.2019.

ЗМІСТ

1.1.	Основні відомості	4
1.2.	Приклад виконання нормалізації відношення	5
1.3.	Індивідуальні завдання	8
1.4.	Рекомендована література	12
2.1.	Основні відомості	13
2.2.	Типи зв'язків	14
2.3.	Ступінь участі (модальність).....	15
2.4.	Створення ER-моделі	16
2.5.	Приклад виконання практичного завдання.....	17
2.6.	Індивідуальні завдання	19
2.7.	Рекомендована література	22

ВСТУП

Самостійна робота студентів є складовою процесу підготовки фахівців зазначених спеціальностей. Самостійна робота сприяє поглибленню знань студентів, розвитку їх творчих здібностей, дозволяє сформувати навички самостійного засвоєння нових знань.

Методичні вказівки визначають базові модулі дисципліни „Організація баз даних”, яка вивчається студентами спеціальності 123 “Комп’ютерна інженерія” усіх форм навчання. В них докладно пояснюється методика рішення задач з конкретної теми дисципліни, подаються посилання на джерела, в яких більш докладно розкриваються теоретичні питання. Методичні вказівки містять індивідуальні завдання та питання для самоперевірки для кожного модулю дисципліни..

1. НОРМАЛІЗАЦІЯ ВІДНОСИН

1.1 Основні відомості

Нормалізація - це процес розділення вихідного відношення на декілька відношень з меншою збитковістю інформації, що дозволяє запобігти аномаліям вставки, вилучення та оновлення записів відношення.

Нормалізація проводиться у наступній послідовності:

- 1.Визначаються повторювальні групи, тобто визначаються атрибути, які можуть містити однакові значення.
- 2.Відношення приводиться у відповідність до 1НФ: повторювальні групи виділяються в окреме відношення, у кожному обирається первинний ключ.
- 3.Визначаються стержневі сутності.
- 4.Визначаються існуючі функціональні залежності між атрибутами вихідного відношення.
- 5.Вибирається первинний ключ відношення.
- 6.Проводиться аналіз функціональних залежностей.
- 7.Відношення приводиться до відповідності 2НФ: всі атрибути з частковою залежністю від первинного ключа виносяться в окремі відношення, у яких обирається первинний ключ.

8.Відношення приводиться до відповідності 3НФ: всі неключові функціонально залежні атрибути виносяться в окремі відношення, у яких обирається первинний ключ.

9.Перевіряється відповідність відношення НФБК: всі детермінанти функціональних залежностей є ключами відповідних відношень.

10.Всі створені відношення перевіряються на відповідність 1НФ, 2НФ, 3НФ та НФБК.

1.2 Приклад виконання нормалізації відношення

Нормалізувати відношення Продаж ліків

Код_ліків	Ліки	Термін придатності	Ціна	Кількість	Аптека	Постачальник

Для нормалізації відношення вихідне відношення приводять до відповідності нормальним формам 1НФ, 2НФ, 3НФ, НФБК

Для приведення вихідного відношення до 1НФ необхідно визначити атрибути, у яких можуть повторюватись значення.

Тому що у всіх аптеках можуть продаються будь-які ліки, то повторювальні групи складають атрибути Найменування ліків та Аптека. Як що припустити, що кожен Постачальник забезпечує всі Аптеки будь-якими ліками, то значення атрибуту Постачальник також повторюються. Тому для приведення вихідного відношення у відповідність до 1НФ необхідно повторювальні групи винести в окремі відношення, а у вихідному відношенні залишити копії ключів створених відношень.

Продаж ліків

Код_ліків	Термін придатності	Ціна	Кількість	Код_Аптеки	Код_Постачальника

Нові відношення Ліки, Аптека, Постачальник можна представити як у табличній формі так і у вигляді схеми.

Нові відношення Ліки, Аптека, Постачальник у табличній формі

Ліки

Код_ліків	Найменування ліків

Аптека

Код_Аптеки	Аптека

Постачальник

Код_Постачальника	Постачальник

Відношення Ліки, Аптека та Постачальник у вигляді схеми
 Ліки(Код_ліків, Ліки)
 Аптека (Код_Аптеки, Аптека)
 Постачальник (Код_Постачальника, Постачальник)

У відношенні Ліки у якості ключів виберемо атрибут Код_ліків, у відношенні Аптека – Код_аптеки, у відношенні Постачальник – Код_постачальника. Тому що створені відношення складаються з двох атрибутів, один з яких є ключовим, то подальша нормалізація цих відношень не потрібна, тому що вони відповідають вимогам всіх нормальних форм.

Проаналізуємо відношення Продаж ліків на наявність функціональних залежностей. Припустимо, що у всіх Аптеках ліки продаються за одною ціною, яку визначає Постачальник. Тоді у вихідному відношенні можна визначити наступні функціональні залежності:

Детермінант	Залежні атрибути	Вид залежності
Код_ліків, код_аптеки	Кількість	Первинний ключ
Код_ліків	Ціна, Термін придатності	Часткова залежність: Ціна та Термін придатності не залежать від Аптеки
Код_Постачальника	Ціна	Транзитивна залежність

Детермінант	Залежні атрибути	Вид залежності
		атрибуту Ціна від неключового атрибуту Код_Постачальника
Код_ліків Код_Постачальника	Ціна	Потенційний ключ

Для відповідності відношення Продаж ліків 2 НФ необхідно усунути часткову залежність: перенести залежні атрибути в окреме відношення, а в вихідному залишити копії ключів. Атрибут Ціна тільки частково залежить від атрибуту Код_ліків, тому цей атрибут необхідно винести в окреме відношення.

Для відповідності відношення Продаж ліків 3 НФ необхідно усунути транзитивну залежність: перенести залежні атрибути в окреме відношення, а в вихідному залишити копії ключів. Атрибут Ціна повністю залежить від атрибутів Код_ліків, Код_Постачальника тому всі ці атрибути необхідно винести в окреме відношення. Атрибути Код_ліків, Код_Постачальника – це складний ключ відношення Постачання ліків.

Постачання ліків

Код_ліків	Код_Постачальника	Ціна

Таким чином у вихідному відношенні Продаж ліків залишаються атрибути. У різних Аптеках може знаходитись різна кількість кожних ліків, а Термін придатності не залежить від Постачальника, тому, що один Постачальник може забезпечити будь-яку Аптеку однаковими ліками декілька разів.

Продаж ліків

Код_ліків	Код_Аптеки	Кількість

Таким чином схема нормалізованих відношень має вигляд:

Постачальник (Код_Постачальника, Постачальник)
 Ліки (Код_ліків, Найменування_ліків, Термін придатності)
 Аптека (Код_Аптеки, Аптека)
 Постачання ліків (Код_ліків, Код_Постачальника, Ціна)
 Продаж ліків (Код_ліків, Код_Аптеки, Кількість)
 У кожному відношенні підкреслені ключові атрибути.

Для перевірки правильності проведення нормалізації необхідно визначити функціональні залежності у кожному відношенні та перевірити, чи є ключами цих відносин детермінант функціональних залежностей. Аналіз всіх відношень показує, що ця умова виконується

1.3 Індивідуальні завдання

Варіант 1. Розклад

Нормалізувати базове відношення та визначити обмеження цілісності

Викладач	День тижня	Пара	Дисципліна	Група

Варіант 2. Аптека

Нормалізувати базове відношення та визначити обмеження цілісності

Група ліків	Ліки	Придатний до	Ціна	кількість	Аптека	Постачальник

Варіант 3. Розклад руху автобусів

Нормалізувати базове відношення та визначити обмеження цілісності

Код рейсу	Маршрут	Модель автобуса	Вартість проїзду	День тижня	Час відправлення

Варіант 4. Ремонт квартир

Нормалізувати базове відношення та визначити обмеження цілісності

Дата	Рег №	Замовник	Адреса замовника	Вид робіт	Вартість роботи	Таб № виконавця	Виконавець	Спеціальність

Варіант 5. Страхова компанія

Нормалізувати базове відношення та визначити обмеження цілісності

№	Дата	Філія	Адреса філії	Телефон філії	Вид страхування	Тарифна ставка страхування	Клієнт	Агент

Варіант 6. Готель

Нормалізувати базове відношення та визначити обмеження цілісності

Дата	Номер кімнати	Клас кімнати	Початок оренди	Кінець оренди	Вартість оренди за добу	Клієнт

Варіант 7. Облік перевезення вантажу

Дата початку поїздки	Дата кінця поїздки	Маршрут	Відстань	Кількість днів	Вартість перевезень	Водій	Стаж	Премія

Варіант 8. Змагання

Нормалізувати базове відношення та визначити обмеження цілісності

Код змагання	Вид змагання	Дата	День тижня	Місто проведення	Учасник	Результат

Варіант 9. Наукова конференція

Нормалізувати базове відношення та визначити обмеження цілісності

№ п/п	Місто проведення	Дата	Тема конференції	Учасник	Статус учасника	Тривалість виступу	Дата виступу
					доповідач		
					організатор		
					слухач		

Варіант 10. Ремонт верстатів

Нормалізувати базове відношення та скласти схеми відношень та визначити обмеження цілісності

№ п/п	Вид верстатів	Рік випуску	Модель	Вид ремонт	Дата початку ремонту	Розміщення по цехах	Кількість у цеху

Варіант 11. Платна поліклініка

Нормалізувати базове відношення та визначити обмеження цілісності

№ запису	Дата	Спеціальність лікаря	ПІБ лікаря	ПІБ хворого	Діагноз	Вартість лікування

Варіант 12. Ломбард

Нормалізувати базове відношення та визначити обмеження цілісності

№ п/п	Клієнт	Категорія товару	Опис товару	Дата прийому	Дата повернення	Сума оцінки	комісійні

* вважаємо, що книга видається на 10 днів

** вважаємо, що у книги один автор, а не колектив авторів

Варіант 13. Бюро по працевлаштуванню

Нормалізувати базове відношення та визначити обмеження цілісності

Роботодавець	Вид діяльності	Телефон роботодавця	Претенденти	Спеціальність	Дата	Комісійні	Заробітна платня
	Виробництво						
	торгівля						

Варіант 14. Ювелірна майстерня

Нормалізувати базове відношення та визначити обмеження цілісності

№ п/п	Клієнт	Виріб	Метал	Проба	Робота	Вартість роботи	Дата виконання

Варіант 15. Ремонт верстатів

Нормалізувати базове відношення та скласти схеми відношень та визначити обмеження цілісності

№ п/п	Вид верстатів	Рік випуску	Модель	Вид ремонт	Дата початку ремонту	Розміщення по цехах	Кількість у цеху

Варіант 16. Відділ кадрів

Нормалізувати базове відношення та визначити обмеження цілісності

Таб. №	Дата при- йому	ПІБ	Дата на- родження	Освіта	ПІБ дітей	Дата на- родження дітей	Дом теле- фон	відділ	Посада	Оклад

Варіант 17. Ремонт верстатів

Нормалізувати базове відношення та скласти схеми відношень та ви-
значити обмеження цілісності

№ п/п	Вид верс- татів	Рік випус- ку	Мо- дель	Вид ремонт	Дата почат- ку ре- монту	Розмі- щення по цехах	Кіль- кість у цеху

1.4 Рекомендована література

<http://kafinfor.petrso.ru/mysql/index.htm>

<http://www.site-do.ru/db/sql3.php>

2. ІНФОЛОГІЧНЕ МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Основні відомості

Інфологічне моделювання предметної області передбачає створення логічної моделі предметної області, яка не залежить від програмного забезпечення, що використовується для побудови додатку у середовищі конкретної СУБД. Частіше використовується модель Сутність-Зв'язок, яка відображає об'єкти, інформація про які зберігається у БД та їх взаємодію. Основними елементами такої моделі є сутність, атрибут, домен, ключ тощо.

Суть – це клас однотипних об'єктів, інформація про яких має бути врахована в моделі, тобто будь-який помітний об'єкт, інформацію про яке необхідно зберігати в базі даних. Сутністю можуть бути люди, місця, літаки, рейси, смак, колір і т.д. Прикладами сутності можуть бути такі класи об'єктів як "Постачальник", "Співробітник", "Накладна".

На ER- діаграмах сутності зображуються у вигляді прямокутників.

Атрибут – поименована характеристика сутності, властивість типу сутності або типу зв'язку. Атрибути сутності містять значення, що описують кожну сутність.

Найменування атрибуту повинне бути унікальним для конкретного типу сутності, але може бути однаковим для різного типу

Прикладами атрибутів сутності "Співробітник" можуть бути такі атрибути як "Табельний номер", "Прізвище", "Ім'я", "По батькові", "Посада", "Зарплата" і т.п

Домен атрибуту - безліч значень, які можуть бути привласнені атрибуту.

Ключ – мінімальний набір атрибутів, по значеннях яких можна однозначно знайти необхідний екземпляр сутності. Мінімальність означає, що виключення з набору будь-якого атрибуту не дозволяє ідентифікувати сутність по атрибутах, що залишилися.

Зв'язок – асоціація двох або більш сутностей. Одна сутність може бути пов'язана з іншою сутністю або сама з собою. Зв'язок показує, яким чином сутності співвідносяться або взаємодіють між собою.

Класифікація сутностей

К.Дейт визначає три основні класи сутностей: стрижньові, асоціативні і характеристичні, та підклас асоціативної суті - позначення.

Стрижньова суть (стрижень) - це незалежна суть. Ця суть існує незалежно від яких або іншої суті. Їх атрибути визначатимуть іншу суть.

Асоціативна сутність - це сутність, що об'єднує властивості деяких інших сутностей, реалізовує зв'язок між двома або більше сутностями або екземплярами сутності. Наприклад, для сутності Іспит атрибути залежать від сильної сутності Викладач, Студент і Дисципліна.

Характеристична сутність (характеристика) - це зв'язок типа М:1 або 1:1 між двома сутностями (окремий випадок асоціації). Єдиною метою характеристики в рамках аналізованої наочної області є опис або уточнення деякій іншій сутності. Існування характеристики цілком залежить від сутності, яка характеризується.

Позначення - це зв'язок типа М:1 або 1:1 між двома сутностями і відрізняється від характеристики тим, що не залежить від сутності, яку він позначає.

Позначення і характеристики не є цілком незалежною сутністю, оскільки вони припускають наявність деякій іншій сутності, яка "описуватиметься" або "характеризуватиметься". Проте, вони є окремими випадками сутності і можуть мати властивості, можуть брати участь в асоціаціях, позначеннях і мати свої власні характеристики. Проте, всі екземпляри характеристики повинні бути обов'язково пов'язані з яким-небудь екземпляром сутності, яка характеризується.

2.2 Типи зв'язків

Зв'язок - осмислена асоціація між сутностями різних типів, що включає по одній сутності з кожного типу сутності, що бере участь в зв'язку.

Кожному типу зв'язку привласнюється ім'я, яке повинне описувати його функцію. Зв'язок зазвичай виражається дієсловом. Студент ВІВЧАЄ Дисципліну, Співробітники ПРАЦЮЮТЬ У Відділах. При цьому ім'я зв'язку повинне бути унікальним в межах БД.

Зв'язок, який сполучає дві сутності, також може мати атрибути. Зв'язок між сутностями може бути обов'язковим або необов'язковим.

Графічно зв'язок зображується у вигляді лінії, що сполучає дві сутності.

На сутності накладаються деякі обмеження, які відображають вимоги реального миру. Наприклад, кожен об'єкт нерухомості має свого власника, на кожному підприємстві є хоч би один співробітник.

Зв'язок Один-до-Одного (1:1): у кожен момент часу кожному екземпляру сутності А відповідає 1 або 0 екземплярів сутності В.

Наприклад: кожен викладач може читати тільки одну дисципліну, і кожна дисципліна може читатися тільки одним викладачем.

Зв'язок типа Один-до-Багатьох означає, що один екземпляр першої сутності (лівою) пов'язаний з декількома екземплярами другої сутності (правою).

Наприклад: кожен викладач може читати декілька дисциплін, і кожна дисципліна може читатися тільки одним викладачем.

Зв'язок типа Багато-до_Багатьох означає, що кожен екземпляр першої сутності може бути пов'язаний з декількома екземплярами другої сутності, і кожен екземпляр другої сутності може бути пов'язаний з декількома екземплярами першої сутності. У реляційній моделі цей тип зв'язку повинен бути замінений двома зв'язками типу Багато-до_Багатьох шляхом створення проміжної сутності.

Наприклад: кожен викладач може читати декілька дисциплін, і кожна дисципліна може читатися декількома викладачами.

2.3 Ступінь участі (модальність)

Необов'язковий зв'язок виражається дієсловом може (необов'язковий зв'язок), а обов'язкова - винен (обов'язковий зв'язок). Модальність "може" означає, що екземпляр однієї сутності може бути пов'язаний з одним або декількома екземплярами іншої сутності, а може бути і не пов'язаний ні з одним екземпляром., тобто не кожен екземпляр сутності повинен брати участь в зв'язку.

Модальність "винен" означає, що екземпляр однієї сутності зобов'язаний бути зв'язаний не менше чим з одним екземпляром іншої сутності, тобто кожен екземпляр даної сутності повинен брати участь в зв'язку.

Зв'язок може мати різну модальність з різних кінців. Наприклад, "Співробітник може мати декілька Дітей", "кожну Дитину повинен мати одного Батька", тобто з боку Співробітник зв'язок має необов'язковий характер, а з боку Дитина – обов'язковий. Зліва направо: "кожен співробітник може мати декілька дітей".

Справа наліво: "Кожна дитина зобов'язана належати рівно одно-

му співробітникові".

При побудові БД в середовищі конкретної СУБД модальність зв'язку відображається в тому, чи може атрибут приймати значення 0 або невизначене значення NULL.

2.4 Створення ER-моделі

Концептуальне проектування – створення інформаційної моделі наочної області, не залежної від яких-небудь фізичних умов реалізації.

При проектуванні бази даних з використанням моделі "сутність-зв'язок" виконуються наступні етапи:

Визначення типів сутності

На підставі аналізу наочної області складається список сутностей, який може змінюватися в процесі проектування. Як правило, сутність присутня в специфікації користувача у вигляді іменників. Слід проаналізувати, які сутності будуть стрижньовими, а які асоціативними.

Визначення ключів сутності.

Визначення атрибуту, який однозначно визначить конкретний екземпляр сутності. Як правило, цей атрибут простий, є послідовність з кроком 1, оскільки ключ сутності є зовнішнім ключем асоціативної сутності, що реалізовує зв'язок між декількома незалежними сутностями.

Визначення типів зв'язків

Встановлюється наявність зв'язків між сутностями, при цьому враховуються тільки ті зв'язки, які задовольняють опису предметної області. В більшості випадків зв'язки є парними, тобто зв'язують дві сутності. Зв'язки присутні в специфікаціях у вигляді дієслів. Необхідно визначити до якого типу зв'язки відносяться встановлені зв'язки (1:1, 1:M, M:K), чи є вони обов'язковими.

Визначення атрибутів і скріплення їх з типами суті і зв'язків

Для кожної сутності виділяються всі властивості, які згадані в специфікації. Ці властивості відбиваються в атрибутах, які описують сутність.

Визначення потенційних і первинних ключів асоціативної суті

Асоціативна сутність реалізує зв'язок між сутностями, тому особливо важливо правильно визначити їх ключі. Як правило, ключі такої сутності є складними, такими, що складаються з декількох атрибутів.

Рішення про виділення класів і підкласів суті.

Для зменшення надмірності інформації часто необхідно виділити сутності в класи і підкласи, оскільки підкласи дозволяють відобразити специфічні властивості окремих екземплярів суті.

2.5 Приклад виконання практичного завдання

ER- модель предметної області створюється на підставі опису предметної області, яка може бути зазначені у словесній формі, у вигляді таблиці тощо. Далі подається приклад створення ER- моделі, яка визначена у табличному вигляді

Побудувати ER- модель відношення Продаж ліків

Код_ліків	Ліки	Термін придатності	Ціна	Кількість	Аптека	Постачальник

1.У вихідному відношенні можна визначити незалежні сутності Ліки, Аптека, Постачальник, тому що їх існування не залежить один від одного.

2.Припустимо, що вартість ліків однакова у всіх аптеках та не залежить від постачальника, тоді атрибут Ціна характеризує сутність Ліки.

3.Атрибути визначених сутностей

Ліки⇒ Код_ліків, Ліки (найменування ліків), Термін_ придатності, Ціна.

Аптека ⇒ Код_аптеки, Аптека.

Постачальник⇒ Код_постачальника, Постачальник.

4.Для встановлення зв'язків між зазначеними сутностями визначимо

Сутності Ліки та Постачальник: кожен постачальник може поставляти багато ліків, але кожен вид Ліків поставляє конкретний Постачальник, тому між цими сутностями зв'язок типу 1:М. Ключовий атрибут сутності Постачальник входить до атрибутів сутності Ліки як зовнішній ключ. Тому до атрибутів сутності Ліки треба додати ключовий атрибут сутності Постачальник

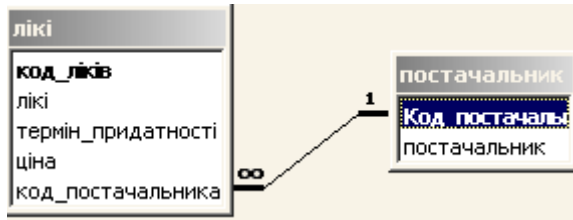


Рисунок 2.12 — Зв'язок 1:M між сутностями Ліки та Постачальник

Сутності Ліки та Аптека: всі Ліки можуть продаватись у будь-якій аптеці та у кожній аптеці продаються багато Ліків, ому зв'язок між цими сутностями типу М:М. У реляційній моделі БД не реалізується зв'язок типу М:М. Такий зв'язок перетворюється у зв'язки типу 1:1 або 1:M шляхом введення додаткової сутності, яка з'єднує зазначені сутності. Введемо сутність Продаж_ліків, яка містить ключові атрибути сутностей Ліки та Аптека у якості зовнішніх ключів, власний атрибут сутності Продаж ліків є атрибут Кількість. У сутності Продаж ліків складний ключ: атрибути Код_ліків та Код_аптеки тому що Кількість Ліків у аптеці визначається Аптекою та видом Ліків.

Тоді ER- модель має вигляд



Рисунок 2.2 — Результуюча ER- модель для предметної області Аптека

2.6 Індивідуальні завдання

Варіант 1. Перукарня

Дата	Вид послуги	Вартість	Пол клієнта	Майстер	Клієнт	Ознака постійного клієнта	Знижка

Варіант 2. Кіноафіша

Код к/фільму	Дата	День тижня	Кінотеатр	К/фільм	Жанр	Тривалість	Сеанси
					Комедія		
					драми		

Варіант 3. Комп'ютерні курси

Група	Дисципліна	Тривалість, тижнів	Кількість слухачів	Вартість	Термін	День тижня	Початок занять

Варіант 4. Видавництво

Видавництво	Жанр	Автор	Назва книги	Кількість сторінок	Рік видання
	Підручник				
	Художня				

Варіант 5. Патентне бюро

Регістраційний номер	Вид	Область	Автор	Дата подачі	Дата реєстрації	Розмір винагороди
	патент	Технічний				
	винахід	побутовий				

Варіант 6. Міжміський автобус

Код рейсу	Маршрут	Модель автобуса	Вартість проїзду	День тижня	Час відправлення	Вокзал

Варіант 7. Реєстрація замовлень на ремонт квартир

Дата	Замовник	Адреса замовника	Вид робіт	Вартість роботи	Виконавець	Спеціальність
						столяр
						кафельник

Варіант 8. Перевезення вантажів

Дата	Замовник	Адреса замовника	Маршрут	Відстань км	Вартість за кг/км	Вага	Марка машини	Водій

Варіант 9. Магазин побутової техніки

Дата	Покупець	Адреса покупця	Товар	Кількість	Ціна	% знижки	Продавець

Варіант 10. Хімчистка

Дата	Вид послуги	Тип	Вартість	Дата виконання замовлення	Клієнт	Ознака постійного клієнта	Знижка
	Чистка						
	прасування						
	фарбування						

Варіант 11. Туристична фірма

Код маршруту	Маршрут	Дата відправлення	Вартість	Кількість туристів	Транспорт

Варіант 12. Кінофільми

Фільм	Тривалість	Студія	Відомі актори	Режисер	Жанр	Вартість

Варіант 13. Оренда автомобіля

Дата видачі	Дата повернення	Марка автомобіля	Всього за оренду	Вартість оренди за добу	Клієнт	Телефон клієнта	Адреса клієнта

Варіант 14. Офісні витрати

Дата видачі	Вид витрат	Норма витрат	Відділ	Кількість робітників	Итого видано
	папір				
	олівець				
	лінійка				

Варіант 15. Облік телефонних розмов

Дата розмови	Вид розмов	Абонент	Номер телефону абонента	Тариф денний	Тариф нічний	Місто	Час доби розмови	Кількість хвилин

Вариант 16. Облік перевезення вантажу

Дата начала поїздки	Дата кінця поїздки	Маршрут	Відстань	Кількість днів	Вартість перевезень	Водій	Ста ж	Пре мія

Вариант 17. Розподіл начального навантаження

Викладач	Вчений ступінь	Посада	Оклад	Дисципліна	Кількість годин	Номер групи
	ктн	доцент				
	дтн	професор				

2.7 Рекомендована література

1. Конноли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика.- Москва-Санкт-Петербург- Киев: Вильямс, 2001.-1111с.

2. Дейт К. ДЖ. Введение в системы баз данных.- Киев- Москва: Диалектика, 1998. - 782с.

3. Хансен Г., Хансен В. Базы данных. Разработка и управление. - Москва: Бином, 2000. - 356с.

4. Г. Гарсия-Молина, Дж. Ульман, Дж. Уидом. Системы баз данных. Полный курс.- Москва-Санкт-Петербург -Киев: Вільямс, 2003.- 1082с.

5. Д.Кренке. Теория и практика построения баз данных. Москва: Питер, 2003: - 800с.

6. Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных. Учебное пособие. Нижний Новгород: Изд-во ННГУ, 2004. 217 с..

7. Карпова Т. Базы данных. Модели, разработка, реализация. Санкт-Петербург: Питер, 2011. - 304с.

8. <http://rema44.ru/resurs/study/dbpri/project2.html>

9. <https://habr.com/ru/post/193136/> 1-3

10. <https://habr.com/ru/post/193284/> 4-6

11. <https://habr.com/ru/post/193380/> 7-9