

Міністерство освіти та науки України
Національний Технічний Університет
«Харківський Політехнічний Інститут»

Кафедра програмної інженерії та інформаційних технологій управління

Методичні вказівки до виконання курсової роботи
з дисципліни
«БАЗИ ДАНИХ ТА СХОВИЩА ДАНИХ»
для студентів денної форми навчання
спеціальності 126 «Інформаційні системи та технології»

Електронне видання

ЗАТВЕРДЖЕНО

кафедрою ПІТУ

Протокол № 1 від 29.08.2018 р.

Харків
НТУ «ХПІ»
2018

Методичні вказівки до виконання курсової роботи з дисципліни «БАЗИ ДАНИХ ТА СХОВИЩА ДАНИХ» для студентів денної форми навчання спеціальності 126 «Інформаційні системи та технології» [Електронне видання] / Укладачі Сокол В.Є., Шматко О.В., Гонтар Ю.М. - Харків: НТУ «ХП», 2018. – с.

Укладачі

В.Є. Сокол,
О.В. Шматко ,
Ю.М.Гонтар

Рецензент:

І.П. Гамаюн

Кафедра програмної інженерії та інформаційних технологій
управління

Зміст

1	Загальні відомості про бази даних	4
1.1	Моделі даних.....	4
1.2	Реляційна модель даних.....	8
1.3	Маніпулювання даними в реляційній моделі	9
1.4	Нормалізація відносин	11
2	Структура курсової роботи	13
3	Вимоги до оформлення роботи	15
4	Порядок виконання курсових робіт	16
	РЕКОМЕНДОВАНА ЛІТЕРАТУРА	27
	Додаток А.....	28
	Додаток Б	29
	Додаток Г	31

1 Загальні відомості про бази даних

1.1 Моделі даних

В основі структури баз даних (БД) лежить поняття моделі даних.

Визначення. Під моделлю даних будемо розуміти набір засобів, що описують типи даних, відносини між даними, семантику даних і обмеження на дані.

Стосовно до БД всі моделі можна умовно розбити на 3 групи:

- фізичні моделі даних;
- логічні моделі, засновані на понятті запису;
- логічні моделі, засновані на понятті об'єкта.

Фізичні моделі даних.

Фізичні моделі є низькорівневим поданням даних і визначають способи їх розміщення, методи доступу до даних, техніку індексування. Фізична модель описує дані засобами конкретної системи управління базами даних (СУБД).

Моделі даних, засновані на понятті запису.

Ці моделі описують дані на концептуальному рівні і рівні уявлення і визначають не тільки архітектуру БД, а й дають загальний опис її реалізації.

Основними типами моделей цього класу є:

- реляційна;
- мережева;
- ієрархічна.
- моделі даних, засновані на понятті об'єкта.

Моделі цього типу застосовуються для опису даних на концептуальному рівні і рівні уявлення користувача. Вони дозволяють визначати структуру БД і обмеження на цілісність даних. Існує багато видів моделей цього класу, найбільш часто використовувані з них - це:

- моделі «сутність-відношення»;
- об'єктно-орієнтовані моделі (ООМ).

Модель «сутність-відношення»

Базовими поняттями в цьому типі моделей є поняття сутності і відносини.

Визначення. Сутністю називається об'єкт, який характеризується (що відрізняється від інших сутностей) набором специфічних ознак, які називаються атрибутами.

Приклад.

Ознаки (атрибути) номер рахунку і залишок по рахунку характеризують (визначають / відрізняють) сутність - банківський рахунок.

Визначення. Відношення будемо називати асоціацію (зв'язок) між різними сутностями.

Приклад.

ВідношенняКлієнт_Рахунок (CustAcc) пов'язує клієнта банку і рахунок, яким він володіє.

Визначення. Безліч всіх сутностей і відношень одного типу називається безліччю сутностей і безліччю відносин відповідно.

Одним з важливих обмежень на дані в цій моделі є потужність відображення, яка визначається кількістю сутностей, пов'язаних з іншою сутністю за допомогою безлічі відносин. При цьому можна за допомогою потужності вказати, скільки примірників батьківської сутності пов'язано з сутністю-нащадком.

Приклад.

Сутність «клієнт» пов'язана з сутністю «рахунок» відношенням «Клієнт_Рахунок». Його потужність - «один до багатьох».

Модель «сутність-відношення» і, тим самим, логічна структура БД, може бути представлена графічно за допомогою E / R діаграми. Основними елементами діаграми є:

- прямокутники, що представляють безліч сутностей;
- еліпси, що представляють атрибути;

- ромби, що представляють відносини між множинами сутностей;
- лінії, які пов'язують сутність з її атрибутами і безліч сутностей з відносинами.

Потужність відносини теж може бути виражена графічно за допомогою вказівки точки на кінці лінії, що відноситься до суті-нащадку, і явного написання потужності відносини.

Усередині кожної з цих фігур записується ім'я, відповідне суті, атрибуту або відношенню. Ці діаграми ще називають діаграмами Чена, на ім'я їх автора.

Об'єктно-орієнтована модель.

Так само як і E / R модель, об'єктно-орієнтована модель заснована на наборі об'єктів. Об'єктно-орієнтований підхід до проектування БД пропонує широкий набір типів. Існують базові типи даних: прості числа, дійсні числа, булеві вирази, рядки символів. Крім того, за допомогою конструктора типів можна з базових типів побудувати нові типи:

- структури записів (структури);
- типи множин (масиви, списки і безлічі);
- типи посилань (показчик).

Об'єкт містить значення даного типу (незмінні об'єкти) або змінні, значення яких відносяться до цього ж типу (змінювані об'єкти), а також функції або процедури, які оперують з об'єктами (методи).

Об'єкти, що містять однакові типи значень і, може бути, однакові методи, групуються в класи.

Метод для класу C повинен мати принаймні один аргумент, який є об'єктом класу C, і може мати інші аргументи будь-якого класу.

Приклад.

Розглянемо клас «Безліч цілих чисел». Для нього можна визначити метод об'єднання двох множин або метод повернення логічного значення, що показує, чи є дане безліч порожнім.

В об'єктно-орієнтованій моделі передбачається, що всі об'єкти мають значення, зване ідентичністю об'єкта (OID). Ніякі два об'єкти не можуть мати один і той же OID, і жоден об'єкт не може мати два різних OID. Т.ч., OID - це значення, яке має посилання на конкретний об'єкт.

Переваги об'єктно-орієнтованого підходу до проектування БД:

1. За допомогою багатой системи типів можна оперувати даними в формах більш природних, ніж відносини в інших моделях даних.
2. Спільне або повторне використання прикладних програм і схем БД за допомогою класів і їх ієрархії простіше, ніж в інших моделях.
3. Абстрактні типи даних допомагають запобігти неправильному використанню даних, якщо дозволити доступ до них тільки за допомогою точно спроектованих функцій.

Абстрактні типи даних.

Класи є абстрактними типами даних. Вони інкапсулюють або обмежують доступ до об'єктів даного класу так, що тільки методи, визначені для цього класу, можуть прямим чином змінювати його об'єкти.

Ієрархії класів.

Один клас А може бути підкласом іншого класу В. Тоді клас А успадковує всі властивості класу В, включаючи тип даних і методи. Однак А може мати і додаткові властивості.

Приклад.

Розглянемо клас, об'єктами якого є банківські рахунки.

```
CLASS Account = {accountNo: integer;
                  Balance: real;
                  Owner: REF Customer;
                  }
```

Можна визначити методи для цього класу, наприклад, Deposit (a: Account, m: real), що збільшує баланс для об'єкта а класу Account на величину m.

Клас Account може мати безліч підкласів, наприклад TimeDeposit, для якого можна визначити додатковий метод: Penalty (a: TimeDeposit), що приписує рахунок а підкласу TimeDeposit і нараховує штраф за передчасне зняття грошей з рахунку як функцію від поля DueDate об'єкта, а й поточної системної дати.

1.2 Реляційна модель даних

В останні роки в більшості БД використовуються реляційні моделі даних, і практично всі сучасні СУБД орієнтовані саме на таке подання інформації. Реляційну модель можна представити як особливий метод розгляду даних, який включає як власне дані (у вигляді таблиць), так і способи роботи і маніпуляції з ними (у вигляді зв'язків). Іншими словами, в реляційній БД використовується кілька таблиць, між якими встановлюються зв'язки. Таким чином, інформація, введена в одну таблицю, може бути пов'язана з однією або декількома записами з іншої таблиці.

Двовимірна таблиця в реляційній теорії називається відношенням. Найменуваннями стовпців в таблиці служать імена атрибутів.

Сукупність схем відносин, що використовуються для представлення моделі даних, називається схемою реляційної бази даних (реляційної моделлю даних). Поточні значення відповідних відносин називаються реляційною базою даних.

Між записами двох таблиць (наприклад, таблиць А і В) можуть існувати такі основні види зв'язків:

- "Один до одного" (кожного запису з таблиці А відповідає одна певна запис з таблиці В, наприклад, працівник отримує зарплату на підприємстві, і тільки одну);

- "Один до багатьох" (кожного запису з таблиці А відповідає кілька записів з таблиці В, наприклад, в будинку проживає багато мешканців)

– "Багато до одного" (безлічі записів з таблиці А відповідає одна певна запис з таблиці В, наприклад, кілька студентів навчаються в одній групі);

– "Багато до багатьох" (безлічі записів з таблиці А відповідає безліч записів з таблиці В, наприклад, у кількох студентів заняття ведуть різні викладачі).

1.3 Маніпулювання даними в реляційній моделі

Для маніпулювання даними в реляційній моделі використовуються два формальних апарату:

- реляційна алгебра, заснована на теорії множин;
- реляційне числення, що базується на обчисленні предикатів першого порядку.

Операції, що реалізуються за допомогою засобів зазначених апаратів, мають важливу властивість: вони замкнуті на безлічі відносин. Це означає, що вирази реляційної алгебри і формули реляційного числення визначаються над відносинами реляційних БД і результатом обчислення також є відносини. В результаті будь-який вираз або формула можуть інтерпретуватися як відношення, що дозволяє використовувати їх в інших виразах або формулах.

Реляційна алгебра має велику виразною потужністю, дуже складні запити до бази даних можуть бути виражені за допомогою одного вираження реляційної алгебри.

Вирази реляційної алгебри будуються на основі алгебраїчних операцій (високого рівня), і подібно до того, як інтерпретуються арифметичні і логічні вираження, вираження реляційної алгебри мають процедурну інтерпретацію.

Операції реляційної алгебри визначені на множині відносин і є замкнутими щодо цього безлічі (утворюють алгебру). Будь довільний запит до БД можна представити у вигляді послідовності, складеної з п'яти базових операцій реляційної алгебри.

Вибір (селекція). позначається $\sigma_p(R)$, де R -відношення, а P - предикат (умова вибору). Як умова вибору може використовуватися будь-який арифметичне або логічний вираз. Результатом цієї операції є безліч рядків (кортежів) відносини, компоненти яких задовольняють умові P .

Проекція. позначається $\Pi_{A_1, A_2, \dots, A_m}(R)$, де R -відношення, A_1, A_2, \dots, A_m - імена стовпців. Результатом операції є безліч рядків (кортежів), отримане з усіх рядків відносини R вибором стовпців з іменами A_1, A_2, \dots, A_m . Іншими словами, це операція побудови «вертикального» підмножини, одержуваного шляхом вибору певних атрибутів і виключення інших. Повторювані рядки виключаються.

Декартовий добуток. Позначається $R \times S$, де R - k_1 -арне відношення, а S - k_2 -арне відношення. Результатом цієї операції є безліч кортежів довжини $k_1 + k_2$, де перші k_1 компонентів яких утворюють кортежі, що належать R , а останні k_2 - кортежі, що належать S .

Об'єднання. позначається $R \cup S$, де R і S - відносини. Результатом операції є безліч кортежів, які належать або R , або S , або їм обом. Для операції об'єднання потрібно однакова арність відносин.

Різниця. Позначається $R - S$, де R і S - відносини. Результатом операції є безліч кортежів, що належать або R і не належать S . Для цієї операції потрібно однакова арність відносин.

Поряд з базовими операціями використовуються додаткові операції:

Перетин. позначається $R \cap S$, де R і S - відносини. Результатом операції є безліч кортежів, що належать як R , так і S . Перетин може бути виражено через вже введені операції

$$R \cap S = R - (R - S).$$

З'єднання. позначається $R \otimes S$, де $R = (A_1, A_2, \dots, A_k, B_1, \dots, B_n)$ і $S = (A_1, A_2, \dots, A_k, C_1, \dots, C_m)$, відносини, які мають однакові атрибути. Результатом операції є безліч кортежів в декартовому творі R і S , що задовольняють висловом

$$R \otimes S = \Pi_{B_1, \dots, B_n, A_1, \dots, A_k, C_1, \dots, C_m} (\sigma_{R.A_1=S.A_1 \& R.A_2=S.A_2 \& \dots R.A_k=S.A_k} (R \times S)).$$

1.4 Нормалізація відносин

Важливим поняттям в теорії реляційних БД є нормалізація відносин. Так як реляційна модель передбачає різні варіанти вибору схем відносин, то від правильного вибору схеми в значній мірі буде залежати ефективність функціонування бази даних.

Основними проблемами при проектуванні логічної моделі БД є:

- Дублювання інформації (надмірність).
- Потенційна суперечливість (аномалії оновлення).
- Потенційна можливість втрати відомостей (аномалії видалення).
- Потенційна можливість невключення інформації в базу даних (аномалії включення).

В теорії реляційних баз даних існують формальні методи побудови реляційної моделі баз даних, в якій відсутні надмірність і аномалії оновлення, видалення та включення.

Побудова оптимального варіанта схем відносин (володіє кращими властивостями при операціях включення, модифікації і видалення даних, ніж всі інші набори схем) здійснюється шляхом нормалізації схем відносин, принципи якої можна сформулювати у вигляді наступних основних правил, які використовуються при розробці структури БД.

У кожному полі таблиці повинен знаходитися унікальний вид інформації, тобто в одній і тій же таблиці не повинні знаходитися повторювані поля.

У кожній таблиці повинен бути первинний ключ або унікальний ідентифікатор, який однозначно визначає даний запис серед безлічі записів таблиці.

Кожному значенню первинного ключа повинна відповідати вичерпна інформація про об'єкт таблиці.

Зміна значення будь-якого поля таблиці, що не входить до складу первинного ключа, не повинно впливати на інформацію в інших її полях.

Нормалізація проводиться в кілька етапів. На початковому етапі схема відносин повинна перебувати в першій нормальній формі (1 НФ). Далі ставлення, представлене в першій нормальній формі, послідовно перетворюється в другу і третю нормальну форми. При деяких припущеннях про дані третя нормальна форма є шуканим найкращим варіантом.

Якщо ці припущення не виконуються, то процес нормалізації триває далі, і відносини перетворюються в четверту і п'яту нормальні форми.

2 Структура курсової роботи

Курсова робота виконується на основі завдання. Приклад оформлення титульного аркуша представлений в дод. А.

Робота складається з вступу, трьох основних розділів, висновків та додатків.

У вступі дається коротка характеристика проблемної області курсової роботи, яка визначається завданням, і обґрунтовується вибір технології, яка використовується для її реалізації.

Перший розділ роботи присвячений функціональній організації розробляється розподіленого додатка. Властивості рівня зберігання даних найбільш зручно виразити у формі скрипта для формування схеми даних. Його необхідно супроводити обґрунтуванням і описом включаються в схему елементів, особливо таблиць. В дод. Б наведено приклад оформлення скрипта.

Для вираження властивостей клієнтського рівня додатки найпростіше привести схеми екранних форм, які необхідно забезпечити відповідними описами. Приклад такої схеми наведений у додатку. В.

У наступній частині розділу необхідно на основі термінології, що вводить при описі скрипта формування схеми даних і схем екранних форм для клієнта, визначити і описати функції, реалізовані проміжним рівнем системи.

У другому розділі роботи визначається структурна організація розробляємого додатку на проміжному рівні. Необхідно декомпонувати його на модулі, в якості яких можуть бути EJB-модулі, HTML-, JSP-, JSF-сторінки, сервлети і т.п.

Взаємозв'язку між модулями найпростіше виразити у вигляді графа, подібного до того, що наведено в дод. Г. В розділі треба привести обґрунтування декомпозиції і опис властивостей виявлених модулів.

У третьому розділі роботи наводиться опис способів вирішення основних проблемних моментів, що виникають при складанні програмного

коду модулів. Крім того, необхідно дати рекомендації з тестування додатка, його випробувань і оформлення як готового виробу.

У висновку в формі висновків по пунктам підводяться підсумки виконаної роботи.

У додатках до курсової роботи наводиться вихідний програмний код розробленого програмного додатка. При необхідності в додатки можна включити найбільш важливі частини коду, але при цьому весь вихідний код повинен бути представлений наданому до роботи машинному носію.

3 Вимоги до оформлення роботи

Порядок проходження частин роботи:

- 1) титульний лист (див. дод. А);
- 2) оформлений бланк завдання;
- 3) зміст;
- 4) основні розділи роботи (1, 2, 3, ...);
- 5) додатки (А, Б, В, ...);
- 6) бібліографічний список.

Робота оформляється на аркушах формату А4, які повинні бути міцно скріплені палітуркою.

Для основного тексту роботи рекомендується використовувати шрифт TimesNewRoman 14. Для програмного коду - моншириний (!) шрифт CourierNew 10.

Вихідний програмний код повинен мати зручну для сприйняття структуру і містити достатню кількість коментарів. Як уже згадувалося, його краще за все розмістити в додатках з відповідними заголовками. Безіменних додатків бути не повинно. При перевищенні рекомендованого обсягу роботивихідний код може бути винісен на машинний носій. Останній повинен бути вкладений в конверт, приклеєний до задньої сторони палітурки роботи.

Бібліографічний список - строго обов'язкова частина роботи. При включенні в нього посилань на сторінки в Internet необхідно, щоб вони були конкретними - простого згадки сайту недостатньо. У текст роботи також включаються посилання на позиції з списку використаних джерел (вказується номер позиції в квадратних дужках). У списку не повинно бути позицій, на які немає посилань в тексті.

Рекомендований обсяг роботи 15 - 20 сторінок.

4 Порядок виконання курсових робіт

Курсові роботи проводяться в два етапи.

Етап 1. Проектування інформаційної системи і її реалізація засобами СУБД MS SQL Server

Завдання 1. Побудувати E / R модель (сформувати суті, відносини і зв'язки між ними). Уявити модель у вигляді діаграми Чена.

Завдання 2. Реалізувати для відповідних моделей запити на мові реляційної алгебри.

Для виконання завдання необхідно вибрати модель інформаційної системи зі списку і здійснити послідовність дій відповідно до завдання.

1. Модель «Деканат» повинна містити інформацію про списки груп, нарахованих стипендії, заліках, оцінках, отриманих на іспитах.

2. Модель «Страхова компанія» повинна містити інформацію про клієнтів, договорах страхівки, об'єктах страхівки, страхових випадках.

3. Модель «Поліклініка» повинна містити інформацію про пацієнтів, декотрих, діагнозах.

4. Модель «Відділ кадрів» повинна містити інформацію про співробітників, підрозділах організації, посадах, офіційних документах (паспорт, диплом, військовий квиток і т.п.)

5. Модель «Фільмотека» повинна містити інформацію про кінофільмах, акторів, кіностудіях і контрактах між акторами і кіностудіями.

6. Модель «Пологовий будинок» повинна містити інформацію про народжених дітей, їх матерів, лікарях, які курирують дитину.

7. Модель «Замовлення квитків» повинна містити інформацію про пасажера, квитку, рейси літака.

8. Модель «Склад» повинна містити інформацію про склад, товар, комірник, документі (накладна на прийом товару, акт на відпуск товару).

9. Модель «Підприємство» має містити інформацію про постачальників, вироблених виробих (склад виробу), споживачах продукції.

10. Модель «Видавництво» повинна містити інформацію про авторів, виданнях, видах рукописів.

11. Модель «Планування матеріально-технічного забезпечення організації» повинна містити інформацію про матеріальні цінності, підрозділах організації, заявках, постачальників матеріальних цінностей.

12. Модель «Канцелярія» повинна містити інформацію про типах документів (вхідні, вихідні), власне про документ, адресата і кореспонденті.

13. Модель «Податкова інспекція» повинна містити інформацію про зареєстрованих платників податків, видах податків, нарахованих і перерахованих сумах податків.

14. Модель «Завод» має містити інформацію про постачальників, деталях і проектах.

15. Модель «Театр» повинна містити інформацію про театр, спектаклі, провідних артистів у виставі.

16. Модель «Собачий каталог» повинна містити інформацію про собаку, її батьків, її господарів.

17. Модель «Швейна компанія» повинна містити інформацію про моделі одягу, про використовувані тканинах, про використовувану фурнітурі, про постачальників тканини і фурнітури.

18. Модель «Торгова компанія» повинна містити інформацію про товари, магазинах, постачальників, покупців.

19. Модель «УБДР» (Управління безпеки дорожнього руху) повинна містити інформацію про зареєстрованих автомобілях, їх власників, виданих документах на автомобіль.

20. Модель «Політклуб» повинна містити інформацію про членів клубу (П.І.Б .; громадянство; число, місяць, рік народження, місце народження; стать; паспорт (серія, номер, ким і коли виданий); освіту (що і коли закінчив); робота (найменування організації, посада); членство в громадських організаціях (найменування); участь у представницьких органах держави і місцевого самоврядування (які, в який час, в якості кого), урядів.

нагороди; контактні телефони; адреса проживання.), про заходи клубу, виконанні персональних доручень, членських внески, пожертвування.

До моделі 1:

- отримати повну інформацію про всі зданих іспитах і заліках в обраній групі;
- отримати список всіх студентів, що не здали залік з обраної дисципліни;
- отримати список студентів, що не здали залік більш ніж по 2-х предметів;
- отримати список студентів, які не отримали стипендію;
- отримати список студентів, які отримали ту ж оцінку на іспиті за обраним предмету, що і обраний студент;
- отримати список предметів, за якими отримано однакову кількість заліків;
- отримати список груп, в яких студенти отримали задану кількість оцінок «п'ять» за обраним предмету.

До моделі 2:

- отримати список клієнтів з заданим об'єктом страхівки;
- отримати список клієнтів з заданим об'єктом страхівки і однаковою страховою сумою;
- отримати список всіх договорів, за якими відбулося заданий страховий випадок;
- отримати список всіх страхових випадків, що сталися з дати по дату;
- отримати список клієнтів, у яких хоча б об'єкт страхівки, що і у заданого клієнта;
- для заданого об'єкта страхівки отримати пари (ім'я клієнта і номер договору), за якими стався страховий випадок.

До моделі 3:

- отримати список всіх пацієнтів із заданим діагнозом;
- отримати список пацієнтів, які обслуговуються тим же доктором, що і заданий пацієнт;
- отримати список лікарів- однофамільців з заданим пацієнтом;
- отримати список лікарів, у яких лікуються пацієнти не менше заданої кількості до заданого діагнозом;
- отримати список пацієнтів, що мають ті ж симптоми хвороби, що і у заданого пацієнта;
- для заданого доктора «А» отримати список його пацієнтів, що мають той же діагноз, що й обраний пацієнт, що лікується у лікаря «В».

До моделі 4:

- отримати список всіх співробітників, що працюють в заданому підрозділі;
- отримати список співробітників, що працюють під керівництвом того ж начальника, що і у вибраного співробітника;
- отримати список всіх однофамільців заданого начальника підрозділу;
- для заданого підрозділу отримати пари (П.І.Б. співробітника, № військового квитка), співробітників, що працюють в заданій посаді;
- отримати список всіх підрозділів, в яких працюють співробітники з вченим ступенем (на підставі диплома);
- отримати список співробітників, що складається з трійок (П.І.Б. співробітника, № паспорта, посада), пенсійного віку. (Звернути увагу на те, що у чоловіків і жінок пенсійний вік різний).

До моделі 5:

- отримати список всіх акторів, які знімаються на заданій кіностудії;
- отримати список кінофільмів, в яких знімаються ті ж актори, що і в заданому кінофільмі;
- отримати пари (П.І.Б. актора, № контракту), зайнятих у фільмах, випущених на заданій кіностудії;

- отримати список кіностудій, в яких були зняті фільми по заданій тематиці і в заданому році;
- отримати список акторів, у яких є контракт з кіностудією, розташованої в тому ж місті, в якому проживають ці актори;
- отримати список кінофільмів, в яких заданий актор не знявся жодного разу.

До моделі 6:

- отримати список всіх матерів, у яких народилися дівчатка;
- отримати список матерів - однофамільницею, із заданим лікарем;
- вибрати докторів, у яких пацієнтами є діти, що народилися в той же день, що і у вибраній матері;
- отримати список лікарів, у яких пацієнтами є близнюки;
- отримати список матерів, у яких народилися хлопчики, на певну дату;
- для заданого доктора отримати пари (мати, дитина), у яких підлогу збігається.

До моделі 7:

- отримати список всіх пасажирів заданого рейсу;
- отримати список пасажирів, що вилітають рейсами до заданого міста і мають квиток на задане місце;
- отримати список рейсів, на яких вилітали пасажирів з заданими П.І.Б. і датою вильоту;
- отримати список, що складається з пар (№ рейсу, № місця, на які не продані квитки);
- отримати список пасажирів, що вилітали до заданого міста в заданому інтервалі дат;
- отримати список типів літаків, що літають по тому ж маршруту, що і літак, на якому летів пасажир із заданим номером квитка.

До моделі 8:

- отримати список всіх товарів, які перебувають на всіх складах;
- отримати список комірників, які брали заданий товар;
- отримати пари (назва товару, № акта), відпущені з заданого складу в заданий день товарів;
- отримати список складів, на які було здійснено прийом тих же товарів, що і на заданий склад;
- отримати номери всіх накладних на прийом тих же товарів, що і брав заданий комірник;
- отримати список (найменування товару, номер складу, П.І.Б. комірника), прийнятих за заданою накладною.

До моделі 9:

- отримати список постачальників заданої деталі;
- отримати список всіх виробів, в яке входить задана деталь від заданого постачальника;
- отримати список споживачів виробу, в які входить та сама деталь, що і проводиться заданим постачальником;
- отримати список виробів, у яких постачальник деталей знаходиться в тому ж місті, що і споживач;
- отримати список споживачів виробу, до якого входить один і той же кількість деталей, вироблених різними постачальниками;
- отримати всі номери деталей, які поставляються тільки одним постачальником.

До моделі 10:

- отримати список всіх видань, що випускаються заданим видавництвом;
- отримати список авторів, які публікують заданий вид рукописи в заданому видавництві;

– отримати список видань, випущених в тому ж видавництві, в якому публікувався автор, який проживає в тому ж місті, що і автор із заданою прізвищем;

– отримати список видавництв і авторів, розташованих в одному і тому ж місті;

– отримати список авторів, які опублікували одну і ту ж рукопис в різних видавництвах;

– отримати пари (автор, видавництво) для рукописів, виданих за заданий період часу.

До моделі 11:

– отримати список всіх мат.цінностей (МЦ), необхідних заданому підрозділу;

– отримати список всіх підрозділів, яким потрібно та ж МЦ, що і заданому підрозділу;

– отримати список пар (МЦ, постачальник) таких, у яких вартість МЦ не більш заданої;

– отримати список пар (МЦ, кількість), необхідних підрозділам організації, що знаходяться в тому ж місті, що і заданий підрозділ;

– отримати номери заявок від підрозділів, яким потрібно задана МЦ;

– отримати список постачальників МЦ, що знаходяться в тій же країні, що і підрозділ організації, якій ця МЦ потрібно.

До моделі 12:

– отримати список всіх вихідних документів за період;

– отримати список всіх кореспондентів, яким надійшли листи в той же день, що і заданому кореспонденту;

– отримати номери вхідних документів від кореспондентів, які проживають в тому ж місті, що і адресат;

– отримати пари (номер вхідного документа, номер вихідного), у яких теми однакові;

- отримати список всіх адресатів, яким були направлені листи на ту саму тему, що і заданому адресату;

- отримати список (кореспонденти, номер вхідного листа), відповіді яких повинні бути підготовлені до заданої дати.

До моделі 13:

- отримати список всіх платників податків, які нараховують заданий вид податку;

- отримати список платників податків, переплата по заданому виду податку;

- отримати список податків, за якими відбулася недоплата;

- отримати суми нарахованих податків платниками податків, які перерахували гроші в той же день, що і заданий платник податків;

- отримати список платників податків, які не перерахували гроші до заданої дати;

- отримати список платників податків, у яких нарахована сума за заданим видом податку лежить в заданому діапазоні.

До моделі 14:

- отримати всі номери деталей, що поставляються постачальником із заданого міста;

- отримати номери деталей, що поставляються для всіх проектів, які забезпечуються постачальником з того ж міста, де розміщений проект;

- отримати список всіх постачальників, що поставляють одну і ту ж деталь для всіх проектів;

- отримати список проектів, що використовують принаймні одну деталь, вироблену заданим постачальником;

- отримати список трійок (номер деталі, кількість, постачальник), аналогічних деталей, що поставляються заданим постачальником;

- отримати список проектів, які забезпечуються деталями від одного постачальника.

До моделі 15:

- отримати список провідних артистів всіх театрів;
- отримати список вистав, в яких зайнятий заданий артист;
- отримати список театрів, в яких грають однофамільці;
- отримати пари (назва театру, місто), в якому йдуть ті ж спектаклі, що і спектакль, в якому зайнятий заданий артист;
- отримати список вистав, в яких беруть участь артисти, які живуть в тому ж місті, що і театр;
- отримати пари (назва театру, спектакль), в яких грають артисти з тим же прізвищем, що і автор п'єси.

До моделі 16:

- отримати список собак заданої породи;
- отримати адреси господарів собаки з заданими батьками;
- отримати список господарів собак заданої породи і заданого забарвлення;
- отримати пари (ім'я господарів, телефон) всіх собак жіночої статі, у яких «тато» був тим же самим, що і у заданій собаки;
- отримати список порід собак, господарі яких проживають на одній вулиці;
- отримати список імен собак заданої породи, у яких батьки були чемпіонами в заданому році.

До моделі 17:

- отримати список всіх постачальників, які виробляють задану тканину;
- отримати список моделей одягу, у яких постачальники тканини і фурнітури розташовані в одному місті;
- отримати пари (номер моделі, номер виробу фурнітури), що входять в модель одягу, зшити із заданої тканини;

- отримати список всіх постачальників, що поставляють тканину і фурнітуру для моделей заданого сезону;
- отримати пари (найменування фурнітури, кількість), що входять в моделі заданого сезону і заданої тканини;
- отримати список номерів моделей одягу, зшитих з тканини заданого кольору.

До моделі 18:

- отримати список всіх товарів, що продаються в заданому магазині;
- отримати список товарів, постачальник яких знаходиться в тому ж місті, що і магазин, їх продає;
- отримати список всіх споживачів, які купили товари вартістю не більше заданої;
- отримати список постачальників, які поставили заданий товар в магазин з заданим адресою;
- отримати пари (номер товару, назва постачальника), що продаються через магазини, в яких обслуговується заданий споживач;
- отримати список пар (магазин, адреса), в яких продається заданий товар від заданого постачальника.

До моделі 19:

- отримати список всіх автомобілів, зареєстрованих в заданому році;
- отримати список власників автомобіля заданої марки;
- отримати список номерів свідоцтв про реєстрацію транспортного засобу тієї ж марки і того ж кольору, що і автомобіль, що належить заданому власнику;
- отримати список пар (марка автомобіля, держ.номер), що належать заданому власнику і зареєстрованому не пізніше заданого року;

- отримати список трійок (П.І.Б. власника, марка автомобіля, гос.номер) автомашин, випущених після заданого року і зареєстрованих в заданому відділенні УБДР;

- отримати список адрес власників автомобілів, зареєстрованих в тому ж відділенні УБДРі в тому ж році, що і автомобіль заданого власника.

До моделі 20:

- отримати список членів політклубу за визначеною формою;
- отримати список членів клубу, що беруть участь в заданому виконавчому органі державної влади;
- отримати список членів клубу, що мають ту ж нагороду, що і заданий член клубу;
- визначити загальну суму внесків за заданий період часу;
- отримати список невиконаних доручень і відповідальних за них.

Завдання 3. Реалізація інформаційної системи в MS SQL Server

Створення нової бази даних

Перед створенням бази даних перевірте установки сервера бази даних: мова (кодова сторінка), методика доступу та ін. У всіх варіантах робочий журнал транзакцій розташовувати в тому ж каталозі, де буде розміщена база даних, з тією ж назвою.

Формування запитів до БД

Сформувані запити до побудованої бази даних інформаційної системи відповідно до обраної моделлю і завданням 2 Етапу 1 виконання курсової роботи.

Результати виконаної роботи оформляються у вигляді звіту.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

1. Watson, R. T. (2016). Data management, databases and organizations (6. ed.). John Wiley & Sons.
2. Elmasri, R., & Navathe, S.B. (2016). Fundamentals of Database Systems, 7th edition. New York: Pearson Education.
3. Provost, F. & Fawcett, T. (2013). Data Science for Business. Sebastopol: O'Reilly Media
4. Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. Big Data & Society, 1(1), 2053951714528481.
5. Bob Bryla, Kevin Loney (2013) Oracle Database 12c The Complete Reference. Oracle Press
6. Carlos Coronel, Steven Morris (2014) Database Systems: Design, Implementation, & Management
7. David M. Kroenke, David J. Auer, Scott L. Vandenberg, Robert C. Yoder (2017) Database Concepts (8th Edition) 8th Edition
8. Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, Kenneth C. Lichtendahl (2017) Data Mining for Business Analytics: Concepts, Techniques, and Applications in R. Wiley
9. Ralph Kimball, Margy Ross (2013) The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling
10. Lawrence Corr and Jim Stagnitto (2011) Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to star schema
11. Matteo Golfarelli and Stefano Rizzi (2009) Data Warehouse design: Modern Principles and Methodology
12. Erik Thomsen (2002) OLAP Solutions: Building Multidimensional Information Systems

Допоміжна література

13. Claudia Imhoff, Nicholas Gallempo, Jonathan G. Geiger: Mastering Data Warehouse Design Relational and Dimensional Techniques, Wiley Publishing, 2003.
14. M. Tamer Özsu, Patrick Valduriez Principles of Distributed Database Systems 3rd ed. 2011 Edition
15. W. H. Inmon Building the Data Warehouse 4th Edition
16. Maarten van Steen, Andrew S. Tanenbaum Distributed Systems 3.01 Edition

Додаток А

Приклад оформлення титульного аркушу курсової роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИНАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»Інститут (факультет) Комп'ютерних наук і програмної інженеріїКафедра Програмної інженерії та інформаційних технологій управлінняСпеціальність _____Спеціалізація _____

КУРСОВИЙ ПРОЕКТ

з дисципліни «Бази даних»

Виконав студент __ курсу, групи _____

(підпис, прізвище та ініціали)

Перевірив _____

(підпис, прізвище та ініціали)

Харків 2018_

Додаток Б

Приклад скрипта для створення схеми даних

```

prompt ===== CreateschemaCollege =====
prompt ----- Droptables
droptableSSRef;
droptableStudent;
droptableSubject;
prompt ----- CreatetableStudent
createtableStudent (
idnumber (6) primarykey,
name varchar2 (16)
);
prompt ----- CreatetableSubject
createtableSubject (
idnumber (6) primarykey,
title varchar2 (16)
);
prompt ----- CreatetableSSRef
createtableSSRef (
idnumber (6),
stnumber (6),
sbnumber (6),
constraint SSRef_pk primarykey (id),
constraint SSRefSd_fk foreignkey (st) referencesStudent,
constraint SSRefSb_fk foreignkey (sb) referencesSubject
);
prompt ----- InsertData
insertintoStudentvalues (101, 'Atos');
insertintoStudentvalues (102, 'Portos');
insertintoStudentvalues (103, 'Aramis');
insertintoSubjectvalues (11, 'Java');
insertintoSubjectvalues (12, 'Pascal');
insertintoSubjectvalues (13, 'CSharp');
insertintoSubjectvalues (14, 'CPP');
insertintoSSRefvalues (10111, 101, 11);
insertintoSSRefvalues (10112, 101, 12);
insertintoSSRefvalues (10212, 102, 12);
insertintoSSRefvalues (10214, 102, 14);
insertintoSSRefvalues (10311, 103, 11);
14
insertintoSSRefvalues (10312, 103, 12);
insertintoSSRefvalues (10313, 103, 13);
commit;
prompt ----- TestData
select * fromStudent;
select * fromSubject;
select * fromSSRef;
selecta.name, b.titlefromStudent a, Subject b, SSRef c
wherec.st = a.idandc.sb = b.idorderbya.name;
selectb.title, a.namefromStudent a, Subject b, SSRef c
wherec.st = a.idandc.sb = b.idorderbyb.title;

```

Додаток В

Приклади схем екранних форм

BANK: Admin

Id	Name	Balance
101	Bess	4000
102	Betty	10000
103	Betsy	2000
104	Eliza	1600
105	Liz	3000
600	Atos	5000
601	Portos	800

Total Balance: **26400**

Operation

Id: Name:

- Create New Account (Id, Name)
 - Delete Account (Id)
 - Renew Table

[Goto Client.jsp](#)

BANK: Client

Account

Id	Name	Balance
101	Bess	4000

Operation

Enter your Id before execute operation.

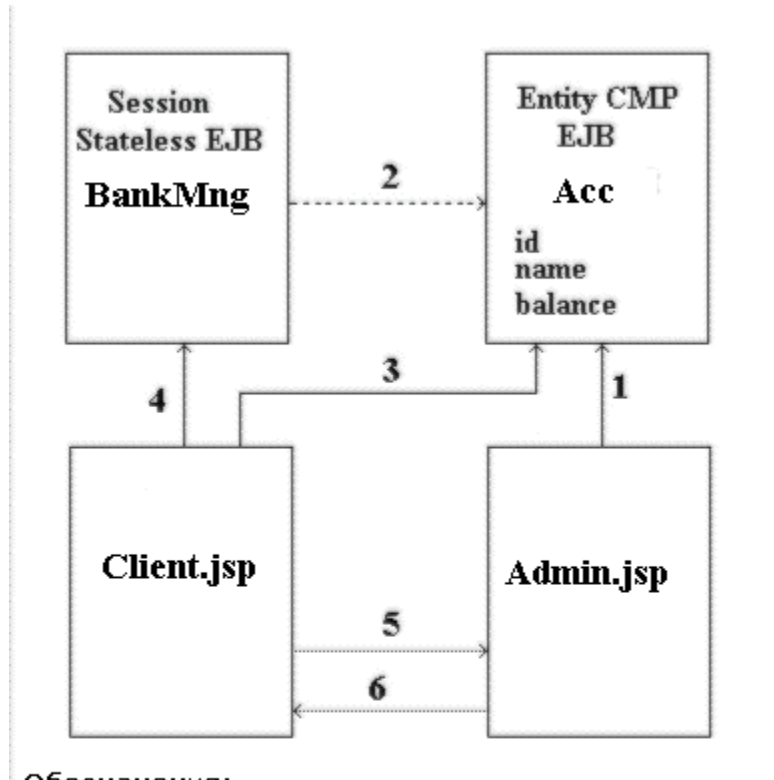
Id: Val:

- Deposit (+Val)
 - Withdraw (-Val)
 - View Data

[Goto Admin.jsp](#)

Додаток Г

Приклад схеми взаємодії модулів



Обозначения:

- > - обращение по локальным интерфейсам
- > - обращение по удаленным интерфейсам
-> - переход по HTML-ссылкам