Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної математики

ЗВІТ

ПРО ВИКОНАННЯ VІІ ЕТАПУ КУРСОВОЇ РОБОТИ

з дисципліни “Бази даних та інформаційні системи”

на тему: Приготування сніданку

Студента ІV курсу, групи КМ-32 напряму підготовки 6.040301 – прикладна математика

ДРАБЧУКА Д.В.

Викладач

ТЕРЕЩЕНКО І.О.

Оцінка: \_\_\_ балів

Київ – 2016

ЗМІСТ

[ВСТУП 3](#_Toc465627279)

[1 ОПИС ЗАВДАННЯ 4](#_Toc465627280)

[2 ОПИС РЕЗУЛЬТАТІВ 5](#_Toc465627281)

[ВИСНОВКИ 13](#_Toc465627282)

[ПЕРЕЛІК ПОСИЛАНЬ 14](#_Toc465627283)

[Додаток А 15](#_Toc465627284)

# ВСТУП

Аналіз бізнес-процесу організаційної системи є важливим для побудови коректної інформаційної системи. Одним із способів такого аналізу є аналіз сутностей та відношень між сутностями шляхом побудови ER-діаграми. Цей спосіб допомагає визначити які відношення між собою мають об’єкти інформаційної системи.

Інформаційна система – це комплекс інформаційних технологій, що реалізують бізнес процеси.

В рамках виконання даного етапу курсової роботи потрібно провести аналіз бізнес-процесу «Здача заліку» шляхом побудови діаграм ERD.

# 1 ОПИС ЗАВДАННЯ

Завдання - описати бізнес-процес «Здача заліку», а саме, побудувати ERD (достатня кількість сутностей – 5 сутностей).

# 2 ОПИС РЕЗУЛЬТАТІВ

Виходячи з завдання, було побудовано діаграму «сутність – зв’язок» (ERD). Діаграма складається з 31 сутності.

Зв’язок «один до багатьох» між будь-якими двома сутностями на діаграмі окрім того, що є за визначенням, означає наступне: ключові поля сутності, яка стоїть зі сторони «один» даного зв’язку, передаються у сутність, яка стоїть зі сторони «багато» в якості зовнішнього ключа і входять у склад ключа сутності, яка стоїть зі сторони «багато». З приводу зв’язків «один до одного» буде пояснення нижче в кожному конкретному випадку з діаграми.

Розглянемо кожну з сутностей.

Користувача системи уособлює сутність User. Вона складається з наступних атрибутів:

а) user\_login – тип variable characters, довжина 10 символів, логін користувача, ключ сутності;

б) user\_pass – тип variable characters, довжина 50 символів, хеш паролю користувача;

в) user\_email – тип variable characters, довжина 30 символів, адреса електронної пошти користувача.

Користувач в системі має певну роль. Ролі зберігаються у сутності Role. Ця сутність має наступні атрибути:

а) role\_name – тип variable characters, довжина 10 символів, назва ролі користувача, ключ сутності;

б) role\_desc – тип text, опис ролі.

Відповідність між ролями та користувачами системи встановлюється за допомогою допоміжної сутності User role, ключ якої складається з ключових атрибутів сутностей User та Role. Більше ніяких атрибутів дана допоміжна сутність не має. Зв’язок між сутностями User та User role є «один до одного», обов’язковий з обох сторін. Цей зв’язок означає, що користувач повинен мати лише одну роль. Зв’язок між сутностями Role та User role є «один до багатьох», обов’язковий з обох сторін. Цей зв’язок означає, що багато користувачів можуть мати одну й ту саму роль, але не може бути такого, щоб якоїсь ролі ні у кого з користувачів не було.

Визначення серед всіх користувачів конкретних викладачів та студентів відбувається наступним чином. В системі є дві сутності: сутність викладачів Teacher та сутність студентів Student. Student має наступні атрибути:

а) student\_name – тип variable characters, довжина 100 символів, ПІБ студента, ключ сутності;

б) student\_nsc – тип variable characters, довжина 10 символів, номер студентської картки, ключ сутності.

Teacher має наступні атрибути:

а) teacher\_name – тип variable characters, довжина 100 символів, ПІБ викладача, ключ сутності;

б) teacher\_nwc – тип variable characters, довжина 10 символів, номер трудової книжки, ключ сутності.

Відповідність між викладачами та користувачами системи забезпечується за допомогою введення допоміжної сутності User-teacher, ключовими атрибутами якої є ключові атрибути сутностей Teacher та User. Більше ніяких атрибутів дана допоміжна сутність не має. Зв’язок між сутностями User та User-teacher є «один до одного», обов’язковий лише зі сторони User-teacher. Цей зв’язок означає, що один викладач обов’язково є одним і лише одним користувачем. Зв’язок між сутностями Teacher та User-teacher є «один до одного», обов’язковий з обох сторін. Це означає, що викладач повинен мати один і лише один обліковий запис.

Відповідність між студентами та користувачами системи забезпечується за допомогою введення допоміжної сутності User-student, ключовими атрибутами якої є ключові атрибути сутностей Student та User. Більше ніяких атрибутів дана допоміжна сутність не має. Зв’язок між сутностями User та User-student є «один до одного», обов’язковий лише зі сторони User-student. Цей зв’язок означає, що один студент обов’язково є одним і лише одним користувачем. Зв’язок між сутностями Student та User- student є «один до одного», обов’язковий з обох сторін. Це означає, що студент повинен мати один і лише один обліковий запис.

Зі студентів складаються такі навчальні одиниці, як навчальні групи. В загальному випадку, група унікально ідентифікується наступним чином. Країна складається з міст. Міста мають університети. Університети складаються (в навчальному розумінні) з факультетів. Факультети складаються з кафедр. Кафедри мають навчальні групи.

Головним атрибутом (ключем) сутності країн State є атрибут state\_name – – тип variable characters, довжина 50 символів, назва країни.

Відповідно до вище зазначених міркувань, сутність міст City зв’язана з сутністю State зв’язком «один до багатьох», обов’язковим з обох сторін. Це означає, що країна обов’язково складається з міст і місто обов’язково входить в склад країни. Головним атрибутом сутності є атрибут назви міста city\_name (тип variable characters, довжина 50 символів). Цей атрибут разом з зовніщнім ключем сутності State утворюють ключ сутності City.

Відповідно до вище зазначених міркувань, сутність університетів University зв’язана з сутністю City зв’язком «один до багатьох», обов’язковим зі сторони сутності University. Це означає, що університет обов’язково має територіальну прив’язку до міста. Атрибути сутності наступні:

а) univ\_abbr – тип variable characters, довжина 10 символів, абревіатура вищого навчального закладу, ключ сутності;

б) univ\_name – тип variable characters, довжина 150 символів, назва вищого навчального закладу.

Перший атрибут разом з зовніщнім ключем сутності City утворюють ключ сутності University.

Відповідно до вище зазначених міркувань, сутність факультетів Faculty зв’язана з сутністю University зв’язком «один до багатьох», обов’язковим з обох сторін. Це означає, що університет обов’язково має хоча б один факультет та факультети мають обов’язкову прив’язку до університету. Атрибути сутності наступні:

а) fac\_abbr – тип variable characters, довжина 10 символів, абревіатура факультету, ключ сутності;

б) fac\_name – тип variable characters, довжина 150 символів, назва факультету.

Перший атрибут разом з зовнішнім ключем сутності University утворюють ключ сутності Faculty.

Відповідно до вище зазначених міркувань, сутність кафедр Department зв’язана з сутністю Faculty зв’язком «один до багатьох», обов’язковим з обох сторін. Це означає, що факультет обов’язково має хоча б одну кафедру та кафедри мають обов’язкову прив’язку до факультету. Атрибути сутності наступні:

а) dep\_abbr – тип variable characters, довжина 10 символів, абревіатура кафедри, ключ сутності;

б) dep\_name – тип variable characters, довжина 150 символів, назва кафедри.

Перший атрибут разом з зовнішнім ключем сутності Faculty утворюють ключ сутності Department.

Відповідно до вище зазначених міркувань, сутність груп Group зв’язана з сутністю Department зв’язком «один до багатьох», обов’язковим зі сторони Department. Це означає, що кафедра, якщо має, то має хоча б одну групу та групи мають обов’язкову прив’язку до кафедри. Атрибути сутності наступні:

а) group\_abbr – тип variable characters, довжина 10 символів, абревіатура групи, ключ сутності;

б) group\_desc – тип text, опис групи.

Перший атрибут разом з зовнішнім ключем сутності Department утворюють ключ сутності Group.

Відповідність між студентом і групою, в якій він навчається, забезпечується введенням додаткової сутності Students and groups, ключовими атрибутами якої є ключові атрибути сутностей Student та Group. Більше ніяких атрибутів дана допоміжна сутність не має. Зв’язок між сутностями Student та Students and groups є «один до одного», обов’язковий з обох сторін. Цей зв’язок означає, що студент обов’язково має (входить в склад) групу та тільки одну. Зв’язок між сутностями Students and groups та Group є «один до багатьох», обов’язковий з обох сторін. Цей зв’язок означає, що група обов’язково має хоча б одного студента.

Навчальні дисципліни (сутність Subject), в загальному випадку, ідентифікуються за шифром (атрибут subject\_number – тип variable characters, довжина 10 символів, шифр дисципліни, ключ сутності), унікальним в рамках університету та, і тому власне, прив’язкою до університету (зовнішній ключ з сутності University). Сутності Subject та University пов’язані зв’язком «один до багатьох», обов’язковим з обох боків. Це означає, що університет обов’язково має хоча б одну дисципліну і дисципліна обов’язково має прив’язку до університету. Ключ сутності University, що є зовнішнім ключем для сутності Subject, разом з атрибутом subject\_number, утворюють ключ сутності Subject. Окрім цього, сутність Subject має наступні атрибути:

а) subject\_name – тип variable characters, довжина 150 символів, назва дисципліни;

б) subject\_ desc – тип text, опис дисципліни.

Кожен викладач може мати різні ролі в рамках тої або іншої дисципліни. Ці ролі зберігаються у сутності ролей викладача Teacher role, яка має наступні атрибути:

а) teacher\_role\_name – тип variable characters, довжина 20 символів, назва ролі вчителя, ключ сутності;

б) teacher\_role\_desc – тип text, опис назва ролі вчителя.

Відповідність між викладачам, групами, ролями викладачів в дисциплінах та дисциплінами встановлюється на основі наступних міркувань. Викладач може в різних групах проводити різні типи занять, відповідно, можуть давати різні типи робіт. Викладач може в різних дисциплінах проводити різні типи занять. Одна й та сама дисципліна може мати різних викладачів з однією роллю, але проводити заняття вони можуть як і в одній групі, так і в різних.

Виходячи з цих міркувань, така відповідність встановлюється введенням допоміжної сутності Subjects, teachers and groups, ключ якої складають ключові поля сутностей Subject, Teacher, Group, Teacher role. Інших атрибутів сутність не має. Всі зазначені сутності пов’язані з допоміжною сутністю зв’язками «один до багатьох» (зі сторони «багато» знаходиться допоміжна сутність), необов’язкових тільки збоку допоміжної сутності (викладач може не мати навантаження заняттями в даний час, група може не мати дисциплін в даний час, може не існувати викладачів конкретної ролі в даний час, дисципліна може не викладатись в даний час, але якщо для будь-якого елементу допоміжної сутності повинні існувати відповідні елементи у сутностях Subject, Teacher, Group, Teacher role). Елемент даної сутності має наступне значення: дисципліну в конкретній групі конкретний викладач може викладати в конкретній ролі

В рамках елементу сутності Subjects, teachers and groups можуть бути створені конкретні роботи (завдання). Це означає наступне: викладач для групи, у відповідності до своєї ролі в рамках дисципліни, може створити роботу (завдання) для виконання студентами групи. Це організовується введенням сутності робіт Work.

Ці роботи є конкретного типу. Типи робіт зберігаються у сутності Work type. Атрибути сутності наступні:

а) work\_type – тип variable characters, довжина 15 символів, тип роботи, ключ сутності;

б) work\_type\_desc – тип text, опис типу роботи.

Правило відповідності між типами робіт та ролями викладачів відображається допоміжною сутністю Teacher role and work type, ключ якої складають ключові поля сутностей Work type та Teacher role. Кожна з цих двох сутностей зв’язана з допоміжною обов’язковим з обох сторін зв’язком «один до багатьох». Пояснюється це наступним чином: кожен тип роботи може розроблюватись обов’язково викладачем хоча б однієї ролі і викладач однієї ролі має розроблювати роботу хоча б одного типу.

Атрибути сутності Work наступні:

а) work\_name – тип variable characters, довжина 30 символів, назва роботи, ключ сутності;

б) work\_desc – тип text, опис роботи;

в) work\_body – тип OLE, тіло роботи.

Перший атрибут сутності Work разом з ключами сутностей Subjects, teachers and groups та Work type (які є, відповідно зовнішніми ключами сутності Work) утворює ключ сутності Work. З даними сутностями сутність Work пов’язана зв’язками «один до багатьох» (сутність Work знаходиться зі сторони «багато»). З сутністю Subjects, teachers and groups сутність Work має обов’язковий з обох сторін зв’язок. Це пояснюється наступним чином: викладач для конкретної групи у рамках дисципліни та у відповідності до ролі у дисципліні повинен мати завдання і, відповідно, завдання повинно мати викладача, який його створив буде перевіряти у цієї групи. З сутністю Work type сутність Work має зв’язок, обов’язковий лише зі сторони Work type. Це пояснюється наступним чином: роботи конкретного типу можуть не існувати в рамках даної дисципліни для конкретної групи, але якщо робота існує, то вона повинна мати тип.

За роботу студенти можуть отримувати оцінку. Це забезпечується за допомогою сутності результатів Result. Атрибути сутності наступні:

а) result\_mark\_value – тип integer, числове значення оцінки;

б) result\_report\_body – тип OLE, тіло звіту з виконаної роботи.

Ключі сутностей Work та Student (які є, відповідно зовнішніми ключами сутності Result) утворює ключ сутності Result. З обома сутностями сутність Mark має обов’язковий з обох боків зв’язок «один до багатьох», що пояснюється наступним чином: кожен студент обов’язково має результат за роботу і кожна робота обов’язково має хоча б одного студента, який її повинен виконати. Елемент сутності

Об’єкти системи (користувач, група, дисципліна, робота, результат, а також кафедра, факультет, університет, місто, країна) може змінювати свій статус. Цей реалізовано за допомогою введення (для кожного з вказаних об’єктів системи) сутностей типу <object> status, які є сутностями статусів об’єктів кожна з таблиць зв’язку «один до багатьох», обов’язкового з обох боків. Цей зв’язок означає наступне: будь-який об’єкт системи має багато статусної інформації, і ця інформація для будь-якого користувача обов’язково повинна мати місце. Сутність типу <object> status має наступні атрибути:

а) <object>\_status – тип variable characters, довжина 10 символів, статус об’єкта, ключ сутності;

б) <object>\_stat\_date - тип date&time, дата зміни статусу об’єкта, ключ сутності;

в) <object>\_stat\_desc – тип text, опис причин зміни статусу об’єкта.

Також до ключа сутності статусів об’єкта входить (в якості зовнішнього ключа) ключ сутності, що уособлює сам об’єкт (тобто, <object> ={User, Group, Subject, Work, Result, Department (or dep), Faculty (or fac), University (or univ), City, State}).

# ВИСНОВКИ

Розроблена в рамках виконання етапу курсової роботи діаграма «сутність-зв’язок» (ERD) для процесу «Здача заліку» складається із:

а) сутностей (кількість – 31);

б) зв’язків.

Основний елемент діаграми ERD - це сутність: множина об’єктів зі спільними характеристиками, які називаються атрибутами. В роботі було використано зв’язки між сутностями наступних типів:

а) один до багатьох – реалізується імпортуванням ключа сутності до сутності, в якій відповідає хоча б двом об’єктам відповідає об’єкт з першої сутності;

б) багато до багатьох - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов’язаних сутностей та з’єднана з ними зв’язком «один до багатьох»;

в) один до одного - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов’язаних сутностей та з’єднана з ними зв’язком «один до багатьох», але для сутності, яка знаходиться зі сторони «багато» реалізуються програмно алгоритми перевірки на відсутність дублікатів об’єктів сутності, яка стоїть зі сторони «один» (тому, фактично сторона «багато» за сенсом є аналогічною стороні «один»).

Окремі зв’язки характеризуються обов’язковістю з боку однієї сутності та необов’язковості з боку другої сутності. Присутні також зв’язки, які характеризуються обов’язковістю з обох боків.

Діаграма ERD, побудована в рамках виконання етапу курсової роботи, відповідає бізнес-логіці, яка була описана в попередніх етапах.

При виконанні домашньої роботи закріплено досвід визначення сутностей та зв’язків між ними і побудови діаграми «сутність-зв’язок» на прикладі процесу «Здача заліку».

# ПЕРЕЛІК ПОСИЛАНЬ

1. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений/Под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2002. – 672с.

2. В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2001.- 496с.

3. Хансен Г., Хансен Д. Базы данных. Разработка и управление. – М.: Бином, 2000. – 704 с.

4. Тимчасове положення про організацію освітнього процесу в НТУУ «КПІ» [Текст] / Уклад.: В. П. Головенкін (розд.: 1-8, 10, 12), С. В. Мельниченко (розд.: 9, 11); за заг. ред. Ю.І. Якименка. – К.: НТУУ «КПІ», 2015. – 102 с.

Додаток А

Діаграма ERD

