

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.9:657

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан факультету

Завідувач кафедри

інформаційних технологій

комп'ютерних наук

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

\_\_\_\_\_ 2021р.

«30» листопада 2021 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему Інформаційно-аналітична система успішності стратегії поведінки на прикладі економічного симулятора

Спеціальність 122 Комп'ютерні науки

(код і назва)

Освітня програма Комп'ютерний еколого-економічний моніторинг

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми**

кандидат технічних наук

(науковий ступінь та вчене звання)

Басараб Р. М.

(підпис)

(ПІБ)

**Керівник магістерської кваліфікаційної роботи**

канд. техн. наук, доцент

(науковий ступінь та вчене звання)

Ткаченко О. М.

(підпис)

(ПІБ)

**Виконала**

(підпис)

Осіпенко Б. С.

(ПІБ студента)

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Голуб Б.Л., к.т.н., доцент

«29» жовтня 2020 р.

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Осипенко Богдана Сергіївна

(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

Освітня програма Комп'ютерний еколого-економічний моніторинг

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Інформаційно-аналітична система успішності стратегії поведінки на прикладі економічного симулятора

затверджена наказом ректора НУБіП України від «29» жовтня 2020 р. №1635 С

Термін подання завершеної роботи на кафедру

Вихідні дані до магістерської кваліфікаційної роботи, інформація про здійснені ходи та результати виконання дій під час проходження симуляції

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Аналіз предметної області.	12.11.2020	Виконано
2.	Моделювання системи.	05.01.2021	Виконано
3.	Інформаційне забезпечення	01.04.2021	Виконано
4.	Розробка системи аналізу даних	20.07.2021	Виконано
5.	Впровадження системи аналізу даних	02.09.2021	Виконано
6.	Попередній захист	30.11.2021	Виконано
7.	Захист	16.12.2021	Виконано

Дата видачі завдання « 29 » жовтня 2020 р.

Керівник магістерської кваліфікаційної роботи

(підпис)

Ткаченко О. М.

(прізвище та ініціали)

Завдання прийняла до виконання

(підпис)

Осипенко Б.С.

(прізвище та ініціали студента)

# НУБіП України

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області та вивчення літератури. Написання 1 розділу, представлення керівнику	до 12.11.2020	Виконано
2	Моделювання системи, погодження з керівником. Написання 2 розділу, представлення керівнику	до 05.01.2021	Виконано
3	Попередній друк 1 та 2 розділу та допоміжних сторінок – титульної, завдання, зміст, вступ, список джерел.	до 01.02.2021	Виконано
4	Опис інформаційного забезпечення.	до 01.04.2021	Виконано
5	Розробка системи аналізу даних: логічна модель бази даних, реалізація сховища даних, аналіз сховища даних.	до 20.07.2021	Виконано
6	Написання 3 та 4 розділу, представлення керівнику	до 20.08.2021	Виконано
7	Впровадження системи аналізу даних. Написання 5 розділу.	до 02.09.2021	Виконано
8	Написання та форматування пояснювальної записки	до 11.11.2021	Виконано
9	Підготовка графічного матеріалу, плакатів, презентації	до 11.11.2021	Виконано
10	Перепліт пояснювальної записки	до 10.12.2021	Виконано
11	Рецензування. Підготовка до захисту	до 15.12.2021	Виконано
12	Захист дипломного проекту	до 16.12.2021	Виконано

**Студентка**

**Керівник магістерської роботи**

(підпис)

(підпис)

**Осипенко Б.С.**

(прізвище та ініціали)

**Ткаченко О.М.**

(прізвище та ініціали)

НУБІП України

## АНОТАЦІЯ

Магістерська кваліфікаційна робота містить 5 розділи.

Основна мета даної роботи створені якісного аналізу для візуального представлення різноманітних стратегій поведінки.

У першому розділі розкритий аналіз предметної області, представлені основні завдання, які має виконувати система, визначена мета.

Другий розділ описує моделювання системи.

У третьому розділі обґрунтовується інформаційне забезпечення.

У четвертому розділі проводиться розробка системи аналізу даних.

У п'ятому розділі відбувається впровадження системи аналізу даних.

## ABSTRACT

The master's qualification work contains 5 sections.

The main purpose of this work is to create a qualitative analysis for the visual representation of various behavioral strategies.

The first section reveals the analysis of the subject area, presents the main tasks to be performed by the system, the defined purpose.

The second section describes system modeling.

The third section substantiates the information support.

The fourth section develops a data analysis system.

The fifth section introduces a data analysis system.

# НУБІП України

## ЗМІСТ

ВСТУП .....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Опис предметної області.....	12
1.2 Огляд інформаційних джерел та існуючих рішень.....	12
1.3 Постановка задачі.....	22
2 МОДЕЛЮВАННЯ СИСТЕМИ.....	23
2.1 Моделювання предметної області.....	23
2.1.1 Діаграма прецедентів.....	24
2.1.2 Діаграма діяльності.....	28
2.1.3 Діаграма послідовності.....	29
3 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	31
3.1 Опис архітектури системи і вузлів.....	31
3.2 Опис вузлів системи, як і поставляють дані по сховищу.....	32
3.3 Загальні поняття з напрямку OLAP-технології.....	32
3.4 Структура бази та сховища даних.....	36
3.5 Збір інформації та її впровадження.....	39
4 РОЗРОБКА СИСТЕМИ АНАЛІЗУ ДАНИХ.....	41
4.1 Механізм вилучення, обробки і передачі даних.....	41
4.1.1 Опис ВІ та створення в його середовищі проекту служби SSAS.....	41
4.1.2 Реалізація отримання даних за допомогою Data Flow.....	45
5 ВПРОВАДЖЕННЯ СИСТЕМИ АНАЛІЗУ ДАНИХ.....	49
5.1 Основні відомості про аналіз даних у середовищах.....	49
5.2 Звіти в середовищі Power BI.....	50
5.3 Розрахунок KPI.....	58
ВИСНОВКИ.....	61
ДОДАТКИ.....	64

НУБІП України

# НУБІП України

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

**BI** - Business intelligence - це метафоричний термін, який не має буквального тлумачення і відноситься до ієрархічно-синергетичного набору автоматизованих інструментів для нетривіального аналізу первинних даних та візуалізації його результатів для підтримки бізнес-аналітики, рішень (підтримка прийняття рішень).

**KPI** - Ключовий показник ефективності - у табличній моделі використовується для вимірювання ефективності значення, визначеного базовим показником, у порівнянні з цільовим значенням, також визначеним показником або абсолютним значенням.

**UML** - Unified Modeling Language - стандартний інструмент для створення «креслень» інформаційних систем (ІС). За допомогою UML можна візуалізувати, деталізувати і уточнювати, конструювати і документувати елементи цих систем.

**БД** - База даних - сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування.

**СД** - Сховище даних - предметно орієнтований, інтегрований, незмінний набір даних, що підтримує хронологію і здатний бути комплексним джерелом достовірної інформації для оперативного аналізу та прийняття рішень.

НУБІП України

НУБІП України

# НУВІП **ВСТУП** України

Актуальність даної роботи полягає в дослідженні перевіреної та отриманої інформації з подальшим порівнянням різноманітних стратегій.

В залежності від різноманітних умов, які можуть статися з людиною та відповідно з її фінансами, на сьогодні доречно проводити дослідження різноманітних стратегій, щоб мати можливість обрати один з можливих варіантів розвитку подій. Наприклад людина може дотримуватися стратегії не використання половини своїх фінансах в різноманітних транзакціях, або навпаки вкладання всіх можливих фінансів в підприємства чи акції.

Дослідження ставить перед нами мету дослідити існуючі системи для візуалізації стратегій та донести покращені розрахунки для кращого впливу та підбору стратегій по яким буде можливість управляти фінансами. Так як, на сьогодні управління фінансами являється актуальним питанням, оскільки

велика кількість людей в період пандемії має в планах збільшити свої фінанси, то намагаються створювати свої малі або великі бізнеси, що в свою чергу може викликати відсутність фінансів на перших етапах створення та реалізації бізнесу. Допомога користувачу розібратися з кредитами та підрахунку різноманітних формул, які допоможуть в розрахунках та веденні бізнесів. За основу взято відому гру Роберта Кійосакі саме даний бізнесмен являється автором бестселеру «Багатий тато, бідний тато» і також паралельно він являється творцем настільних ігор Cashflow 101, які дозволяють підвищити фінансову обізнаність та виділяє 10 порад на рахунок створення та розвитку свого бізнесу[1].

Тому головною метою дослідження є створення інформації за допомогою якої користувач зможе проаналізувати різноманітні стратегії, що дозволить оптимізувати пошуки потрібної стратегії для користувача. Саме дослідження розкриває перед нами можливість вдосконалити свої знання по управлінню персональних фінансів та фінансів підприємства.

**НУВІП України**

**Мета дослідження.**  
Створення якісного аналізу для візуального представлення різноманітних стратегій поведінки.

**Об'єкт дослідження.**

Процес аналізу успішності стратегій поведінки.

**НУВІП України**

**Предмет дослідження.**  
Предметом є інтелектуальна система моніторингу на основі економічного симулятора.

**Методи дослідження.**

**НУВІП України**

Першим етапом в провадженні є вибір рішення даної задачі. А саме:

1. Методи для збору інформації

Даний блок зорієнтований на захоплення інформації з зовнішніх джерел і реалізує збереження, про що йдеться у наступному блоці. За збір інформації може відповідати як людина так і автоматизований модуль. Джерелом є excel файл який заповнюється під час проходження симуляції за допомогою настільної верії даного економічного симулятора.

2. Методи збереження інформації

Другий блок використовує перший блок і зберігає в собі інформації з якою можна буде працювати, а тобто оброблювати. Засобом для збереження є база даних\сховище даних(OLAP система).

3. Методи обробки інформації

Третій блок є модулем для впорядкування інформації її підготовки для аналізу і в подальшому для різних потреб інформування

4. Методи відображення інформації

Четвертий блок є методи для створення та побудови логіки роботи з відображенням та звітами, за потребою.

**НУВІП України**

**Апробація роботи.**



Були опубліковані тези на тему:

1. **МОНІТОРИНГ ТА АНАЛІЗ УСПІШНОСТІ СТРАТЕГІЇ ПОВЕДІНКИ НА ПРИКЛАДІ ЕКОНОМІЧНОГО СИМУЛЯТОРА** тези були представлені в рамках: XI Міжнародній науково-практичній конференції студентів, аспірантів та молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ЕКОНОМІКА, ТЕХНІКА, ОСВІТА»[2].

2. **АНАЛІЗ УСПІШНОСТІ СТРАТЕГІЇ ПОВЕДІНКИ НА ПРИКЛАДІ ЕКОНОМІЧНОГО СИМУЛЯТОРА** тези були представлені в рамках: IV Всеукраїнської науково-практичної інтернет конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ 2021»[3].

**Структура роботи складається з 5 розділів:**

### **1. Аналіз предметної області;**

Розділ «Аналіз предметної області» виконує етап огляну рішень за тематикою дослідження, підіймає питання які інформаційні ресурси варто використати для огляду існуючих рішень. До предметної області входить і визначення інформації яка охарактеризує дослідження роботи та, які завдання і методи використані для її розгортання.

### **2. Моделювання системи;**

Розділ «Моделювання системи» виконує етап системного аналізу з побудовою діаграм які зображають роботу системи та деталізують її на різні етапах і процесах які виконує система в реальному житті.

### **3. Інформаційне забезпечення;**

Розділ «Інформаційне забезпечення» виконує процес побудови архітектури та просування вузлів у інформаційній системі, як побудована система і описує про перехід даних від внесення з зовнішнього джерела даних до збереження їх в базі даних і з перенесенням до сховища даних.

# НУБІП України

## 4. Розробка системи аналізу;

Розділ «Розробка системи аналізу» виконує етап розробки системи з описом усіх модулів які використовує система та їх роботою від розгортання сховища даних до збереження інформації у ній.

# НУБІП України

## 5. Впровадження аналізу даних.

Розділ «Інформаційне забезпечення» виконує головний презентаційний етап в системі. Побудовані звіти відображають аналіз даних що містить різні діаграми, гістограми, моделі відношень даних з числовими характеристиками та їх описом і підрахунком.

# НУБІП України

# НУБІП України

# НУБІП України

# НУБІП України

# НУБІП України

# НАУБІП України

## 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Опис предметної області

Будь яка задача та дослідження розпочинається з того щоб зрозуміти, яка предметна область магістерського дослідження її проаналізувати та подати у зрозумілому вигляді тому коли мова йде про проблематику виразимо те, що предметна область в цій ніші дослідження є в першу чергу взаємодія якості, кількості та доступності інформації для побудови економічних стратегій.

Аналізуючи системи, які на сьогодні створені та впровадженні варто зазначити, що кожна система має як ряд позитивних так і негативних елементів.

### 1.2 Огляд інформаційних джерел та існуючих рішень

На сьогоднішній час існують велика кількість версій економічних симуляторів. Провівши дослідження було вирішено досліджувати дані по принципу гри «Грошовий потік», оскільки програми для даної тематики досить цікаві і в плані реалізації алгоритмів та вображення результатів. Було досліджено online версію гри, яка ґрунтується повністю на паперовій версії гри «Cash Flow» [4]. Дана версії гри реалізована на мові оригіналу, тобто англійській мові. На рис. 1.1. відображено початкове вікно гри.



Рис. 1.1. Початкове вікно гри

Після натискання у додатку користувача переадресують на вибір кімнати для участі в сеансі, що відображено на рис. 1.2.



Рис. 1.2. Вікно для вибору кімнати

Користувач може не тільки підключитися на стіл до інших учасників, а також створити новий стіл, що відображено на рис 1.3. Користувач може обрати кількість учасників, які зможуть приймати участь в даному сеансі симуляції.



Рис 1.3. Вікно для створення нової кімнати

Після початку гри користувач потрапляє на ігрове поле зображене на рис. 1.4. та йому пропонують виорати головну ціль для завершення гри, що відображено на рис. 1.5 оскільки під час в участі на зовнішній частині поля є певні умови для завершення гри, а саме або придбання цілі або зібрати певний розмір доходів.



Рис. 1.4. Ігрове поле



Рис 1.5. Вибір мети для завершення гри

# НУБІП України

Після вибору цілі користувачу надається доступ до вікна з кидком кубика, зображено на рис. 1.6.

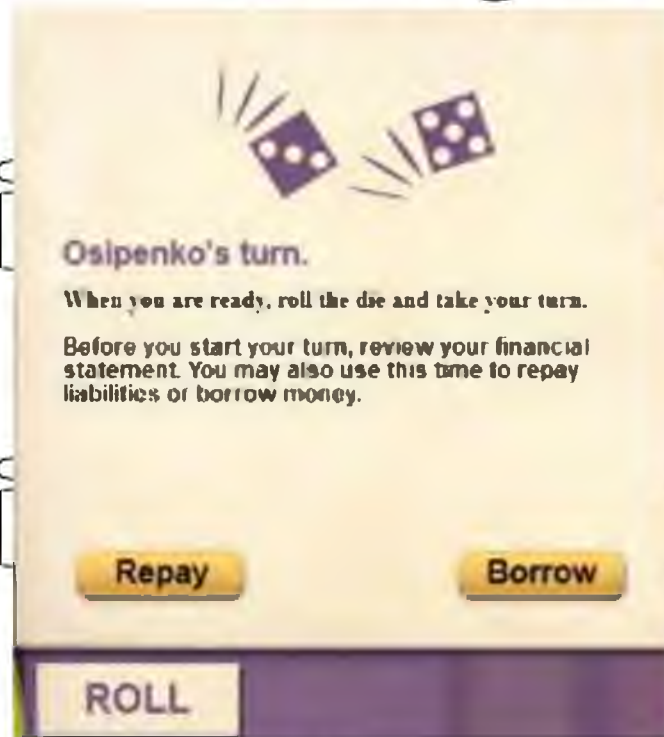


Рис. 1.6. Вікно для кидання кубика

Після кидка кубика гравцю виводять інформацію про поле на яке він потрапив та дію яку він може здійснити. По прикладу зображеному на рис. 1.7 можна побачити що гравцю пропонують зробити благодійний внесок.

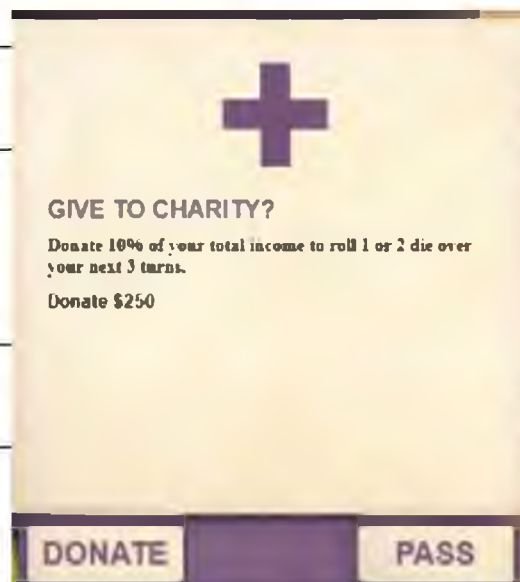


Рис. 1.7. Вікно для підтвердження або відмови від дії

Також невід'ємною частиною даної системи є інформація по фінансах гравця, що зображено на рис. 1.8, а саме:

# НУВІП України

- Розмір заробітної плати;
- Місце, де буде вказана інформація про підприємства, земельні ділянки та житлову площу, яка буде приносити пасивний дохід;

- Витрати гравця;

# НУВІП України

- Акції;
- Кредити
- Інформація про прогрес, оскільки для переходу на зовнішнє коло гравцю потрібен пасивний дохід в 20 разів більший ніж витрати;

- Інформація про наявні кошти, сума прибутків та витрат і щомісячний грошовий потік

# НУВІП України

INCOME		EXPENSES	
	Cash Flow		
Truck Driver Salary	\$2,500	Taxes	\$460
Interest/Dividends		Home Mortgage Payment	\$400
Real Estate/Business		Car Loan Payment	\$80
		Credit Card Payment	\$60
		Retail Payment	\$80
		Other Expenses	\$570

ASSETS		LIABILITIES	
Stocks/Funds/CDs	Cost/Basis	Home Mortgage	\$28,000
		Car Loans	\$4,300
		Credit Cards	\$2,000
Real Estate/Business	Cost	Retail Debt	\$1,000
		Real Estate/Business	Liability

INCOME SUMMARY	
TOTAL INCOME:	\$2,500
TOTAL EXPENSES:	\$-1,620
PAYDAY	\$880

EXPENSES SUMMARY	
TOTAL EXPENSES:	\$1,620

CASH SUMMARY	
CASH:	\$750

INCREASE PASSIVE INCOME TO ESCAPE THE RAT RACE.

PASSIVE INCOME: \$0

Рис. 1.8. Інформація про фінанси гравця

Також було досліджено ще одну версію гри але на цей раз російською мовою відображено на рис. 1.9. В порівнянні з минулою версією, яку розглядали на попередньому кроці в цій версії під назвою CashGo [5] гравець

# НУВІП України

має можливість перегляду мінімальних діаграм, а саме відображення схеми плану для візуалізації гравцю найкращого результату, якого гравець міг досягнути та діаграми з доходами гравця без вирахування витрат (рис. 1.10).

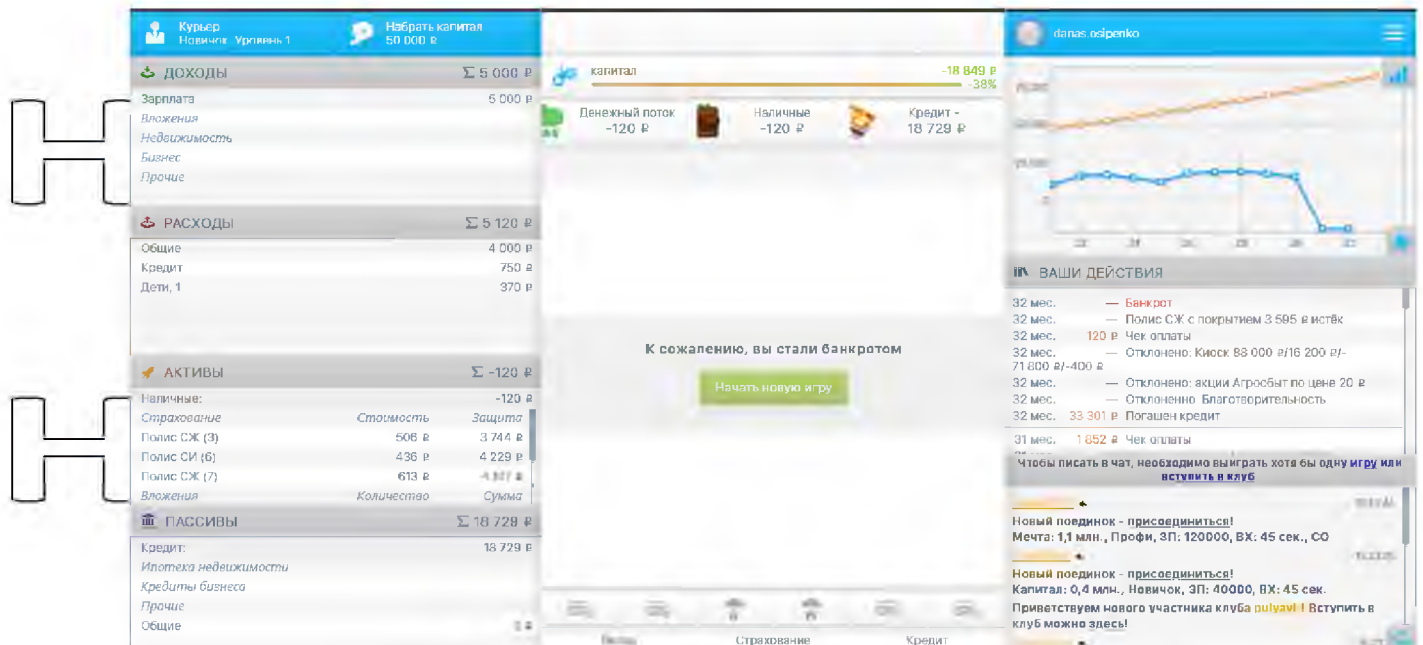


Рис. 1.9 Ігрове поле онлайн версії гри

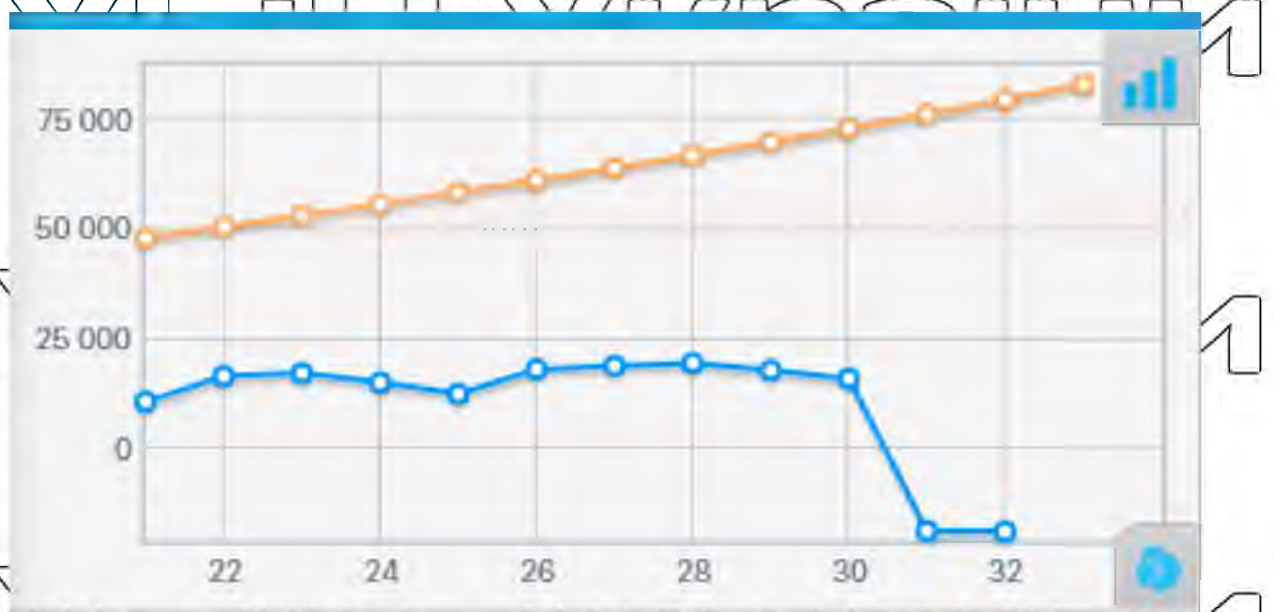


Рис. 1.10 Відображення дій гравця на діаграмі

## 1.2 Інформаційні джерела з предметної області

Для дослідження тематики магістерського дослідження було використано настільну версію гри що відображена на рис. 1.11.

НУВБІП УКРАЇНИ





Рис. 1.11 Настільна версія гри «Грошовий потік»

Для отримання інформації з даної гри була вибрана ця гра оскільки візуальне представлення її клітинок в грі (рис. 1.12) та значення на картках в відповідності від ситуації (рис. 1.13-1.14).



Рис. 1.12 Карта гри



Рис. 1.13 Картки гри



Рис. 1.14 Картки гри

Оскільки дані відображені у вигляді карток то було прийнято рішення сформувати Excel файл в який були внесені дані і який в подальшому був експортований до БД та СД. Для БД було створено таблицю де певним значенням на карті було призначено певний номер, який призначений для спрощення внесення даних рис. 1.15.

	A	B
1	ID_run_condi	ValueD
2	1	Угода
3	2	Неочікувані витрати
4	3	Події на ринку
5	4	Зарплата
6	5	Дитина
7	6	Звільнення
8	7	Повернення на роботу

Рис. 1.15 Таблиця з набором дій

Також була створена таблиця з професіями рис. 1.16, яка призначена для отримання початкових доходів та витрат.

	A	B	C	D	E
1	ID_Prof	Name_Prof	Total_inc	Cumulative_costs	Monthly_stream
2	1	Викладач	170	110	60
3	2	Водій трамвая	250	140	110
4	3	Продавець в магазині	300	120	180
5	4	Прораб	500	230	270
6	5	Хірург	1500	910	590
7	6	Помічник прокурора	2000	1400	600
8	7	Адвокат	2500	1550	950
9	8	Топ-менеджер в міжнародній компанії	3000	1900	1100

Рис. 1.16 Таблиця з інформацією по професіями

Крім цього була створена таблиця з часом та датою для відображення кодів рис. 1.17.

	A	B	C	D	E	
1	ID	Data	TimeD	Weekday	MonthD	YearD
2	1	8:30:00	10	3	2021	
3	2	8:31:00	10	3	2021	
4	3	8:32:00	10	3	2021	
5	4	8:33:00	10	3	2021	
6	5	8:34:00	10	3	2021	
7	6	8:35:00	10	3	2021	
8	7	8:36:00	10	3	2021	
9	8	8:37:00	10	3	2021	
10	9	8:38:00	10	3	2021	
11	10	8:39:00	10	3	2021	
12	11	8:40:00	10	3	2021	
13	12	8:41:00	10	3	2021	
14	13	8:42:00	10	3	2021	
15	14	8:43:00	10	3	2021	
16	15	8:44:00	10	3	2021	
17	16	8:45:00	10	3	2021	
18	17	8:46:00	10	3	2021	
19	18	8:47:00	10	3	2021	
20	19	8:48:00	10	3	2021	
21	20	8:49:00	10	3	2021	
22	21	8:50:00	10	3	2021	
23	22	8:51:00	10	3	2021	
24	23	8:52:00	10	3	2021	
25	24	8:53:00	10	3	2021	
26	25	8:54:00	10	3	2021	
27	26	8:55:00	10	3	2021	
28	27	8:56:00	10	3	2021	
29	28	8:57:00	10	3	2021	
30	29	8:58:00	10	3	2021	
31	30	8:59:00	10	3	2021	
32	31	9:00:00	10	3	2021	
33	32	9:01:00	10	3	2021	
34	33	9:02:00	10	3	2021	
35	34	9:03:00	10	3	2021	
36	35	9:04:00	10	3	2021	

Рис. 1.17 Таблица з інформацією по ходам

Та останньою була створена таблиця з журналом ходів рис. 1.18.

8	217	■	6	0	18950	18950	900	17960	0	0	49840	0
9	218	■	7	170	18950	19120	900	18130	0	0	49840	0
0	219	■	3	170	18950	19120	900	18130	0	0	49840	0
1	220	■	6	0	18950	18950	900	17960	0	0	49840	0
2	221	■	1	0	18950	18950	900	17960	0	0	49840	0
3	222	■	1	0	19100	19100	900	18110	0	15000	34840	70000
4	223	■	3	0	19100	19100	900	18110	0	0	34840	70700
5	224	■	7	170	19100	19270	900	18280	0	0	34840	71400
6	225	■	5	170	19100	19270	1130	18440	1000	0	35840	72100
7	226	■	2	170	19100	19270	1130	18440	0	0	35840	72800
8	227	■	1	170	19100	19270	1130	18440	0	0	35840	73500
9	228	■	5	170	19100	19270	1170	18100	1000	0	36840	74200
0	229	■	4	170	19100	19270	1170	18100	18100	50000	4940	24900
1	230	■	6	0	19100	19100	1170	17930	0	0	4940	25600
2	231	■	7	170	19100	19270	1170	18100	0	0	4940	26300
3	232	■	5	170	19100	19270	1210	18060	1000	0	5940	27000
4	233	■	4	170	19100	19270	1210	18060	18060	0	24000	27700
5	234	■	1	170	19100	19270	1210	18060	0	10	23000	28400
6	235	■	5	170	19100	19270	1250	18020	1000	0	24000	29100
7	236	■	6	0	19100	19100	1250	17850	0	0	24000	29800
8	237	■	4	0	19100	19100	1250	17850	17850	0	42840	30500
9	238	■	4	0	19100	19100	1250	17850	17850	0	60600	31200
0	239	■	7	170	19100	19270	1250	18020	0	11200	29400	0

Рис. 1.18 Таблиця з журналом гри

### 1.3 Постановка задачі

Основне завдання даного магістерського дослідження – це створення якісного аналізу для візуального представлення різноманітних стратегій поведінки в розрізі що для одного гравця, що для декількох:

- 1) Активних фінансів
- 2) Пасивних фінансів
- 3) Грошового потоку
- 4) Кредитів
- 5) Активних фінансів

Для вирішення представлених вище задач, необхідно зібрати набір даних для аналізу залежностей, а саме:

- 1) Нарахування фінансів
- 2) Списання фінансів
- 3) Прирівнення активних фінансів до 0, якщо сталося звільнення
- 4) Повернення значення активних фінансів при прийнятті назад на роботу
- 5) Збільшення витрат в залежності від ситуації

# НУБІП України

## 2. МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1 Моделювання предметної області

UML — це графічне позначення, призначене для опису та моделювання процесів, що відбуваються під час розробки. Схеми цього позначення відрізняються за типом і описують різні аспекти розвитку. Існує 2 основних типи діаграм UML: структурні та поведінкові.

Структурні схеми відображають елементи, з яких складається система.

Поведінкові моделі описують процеси, що відбуваються в системі.

Кожна мова моделювання має свій словниковий запас. Словник UML складається з таких елементів:

**Сутності** - це абстракції, які є основними елементами моделей;

**Відносини** – це зв'язки між сутностями;

Діаграми відображають взаємодію сутностей і відносин.

#### 1. Сутності

Сутності UML можуть бути чотирьох типів:

**Структурні сутності** — це назви істотної моделі, її статичних частин.

Це класи, компоненти, інтерфейси тощо, які відповідають фізичним елементам системи.

**Поведінкові сутності** є керівниками моделі, вони описують її поведінку в часі та просторі. Існує всього 2 види: взаємодія (обмін повідомленнями) і автомат (послідовність станів).

**Об'єкти анотації** є пояснювальними частинами моделі, це примітки та коментарі до елементів системи.

**Об'єкти групування** є організуючими частинами моделі. Вони організують елементи системи в групи.

#### 2. Відносини.

У UML є 4 типи зв'язків:

# НУБІП України

**Залежність** – це відносини між двома сутностями, в яких зміна одного (незалежного) об'єкта призводить до зміни другого об'єкта. Графічно це зображено пунктирною стрілкою.

**Узагальнення** - це відношення "спеціалізація-узагальнення", де спеціалізований об'єкт може бути замінений узагальненим об'єктом.

**Асоціація** – це відношення, яке описує семантичний зв'язок між об'єктами. Графічно він представлений суцільною стрілкою, яка може містити імена персонажів або ролей. (**Агрегація** є різновидом асоціації і відображає відношення частини до цілого).

**Композиція** – це своєрідна агрегація, де взаємозв'язок частини з цілим ще міцніший.

**Виконання** - відношення між кваліфікуючими ознаками, при якому один визначає зобов'язання, а другий досягає його виконання.

### 3. Діаграми.

В UML є 10 типів діаграм:

1. Об'єктів
2. Класів
3. Взаємодії
4. Варіантів використання (use case)
5. Послідовності
6. Станів
7. Кооперацій
8. Компонентів
9. Розгортання
10. Діяльність [6].

#### 2.1.1 Діаграма прецедентів

Дана діаграма дозволяє візуалізувати поведінку системи, підсистеми або класу, щоб користувачі могли зрозуміти, як їх використовувати, а розробники реалізувати відповідний елемент.

НУБІП УКРАЇНИ

Попередньо прецеденти складаються з безлічі сценаріїв (послідовності кроків, описують взаємодію між користувачем і системою), об'єднаних

спільною метою користувача. Відносно складна система містить десятки прецедентів, кожен з яких може розгортатися в десятках сценаріїв. Для будь-

НУБІП УКРАЇНИ

якого прецеденту ми можемо вибрати основні сценарії, які описують найважливіші послідовності, і допоміжні, які описують альтернативні послідовності. Кожен прецедент має передумову і постумову

Є два типи прецедентів: бізнес-прецеденти (описують функціональність на верхньому рівні і призначені для замовника програмної системи) і системні

НУБІП УКРАЇНИ

прецеденти (я описую функціональність на нижньому рівні, призначену для розробника програмної системи). Будь-який прецедент повинен мати

унікальну назву для пакета.

Ім'я може охоплювати кілька рядків. На практиці короткі словесні

речення в активній формі використовуються для назви прецедентів, що

НУБІП УКРАЇНИ

позначають певну поведінку. Прецеденти можна організувати, визначивши між ними відносини узагальнення, розширення та використання.

Відношення узагальнення між прецедентами подібне до відносин узагальнення між класами. Це означає, що попередній нащадок успадковує

НУБІП УКРАЇНИ

поведінку та семантику свого батька, може замінити або доповнити його поведінку, а крім того, може бути замінений скрізь, де з'являється його батько.

Використання відбувається, коли фрагмент поведінки системи повторюється в кількох прецедентах.

Відношення розширення (Extend) по суті схоже на відношення узагальнення. Під час побудови моделі прецедент розширення може

НУБІП УКРАЇНИ

доповнювати поведінку базового прецеденту, але останній повинен визначати точки розширення (тільки для UML). У цьому випадку прецедент розширення

може лише завершити поведінку бази в цих точках розширення. Відношення

НУБІП УКРАЇНИ

розширення використовується для більш точного опису зміни нормальної поведінки.



Слід зазначити, що словесний опис будь-якого прецеденту, навіть самого елементарного, повинен включати щонайменше 3 розділи передумови, основного потоку, постумови. Однак у загальному випадку опис попереднього може включати один або кілька альтернативних потоків, пов'язаних з різними

етапами основного потоку, посилання на інші попередні, вказівку конкретних акторів, задіяних у попередньому (розділ «Актори / актори») [7].

Діаграма прецедентів для системи аналізу успішності стратегії поведінки з позиції активності користувачів наведена на рис. 2.1

Проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

Діаграма прецедентів для системи аналізу успішності стратегії поведінки з позиції активності користувачів наведена на рис. 2.1

Проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.



Рис. 2.1 Діаграма прецедентів

З діаграми видно, що з системою працюють чотири основні актори: гравець, аналітик, системний адміністратор та підприємець. Гравець працює з наповненням даних в систему, задача аналітика опрацювати ці дані, створити

аналітику та донести свої результати до підприємця. В свою чергу, системний адміністратор має підтримувати роботу системи.

Кожен з них має свої варіанти використання, зробимо їх детальний опис в таблиці 1.

Таблиця 1. Опис прецедентів

Основний актор	Назва прецеденту	Опис прецеденту
Гравець	Здійснення ходів	Гравець за допомогою кидка кубика зміщується по карті та здійснює хід у відповідності до того, що випало йому на ігровому полі
	Операції над фінансами	Отримавши карту гравець може виконати операцію відповідно до карти, яка йому випала, що призведе до нового запису в журналі
Системний адміністратор	Моніторинг за коректністю роботи системи	Контроль роботи системи
	Оновлення даних	Оновлення даних про гравців та професії
Аналітик	Підготовка результатів Аналіз результатів дій гравців	Обробка даних отриманих зі сховища для надання інформації підприємцю Проведення аналізу отриманої інформації для подальшого надання її підприємцю
	Формування КРІ	Обрахунок ключового показника ефективності

Таблиця 1. Опис процесів (продовження)

Аналітик	Формування КРІ	Обрахунок ключового показника ефективності
Підприємець	Перегляд звітів Ухвалення рішень про ефективність стратегії	Дослідження результату аналізу аналітика На основі отриманих результатів прийняття або ні результату стратегії

### 2.1.2 Діаграма діяльності

В уніфікованій мові моделювання (UML) діаграма діяльності являє собою графічне представлення всіх дій, які виконує процедурна система, і враховується зміна діаграми стану. Діаграми діяльності детально описують паралельні та умовні дії, варіанти використання та системні функції.

Діаграма діяльності використовується для моделювання послідовного робочого процесу великої діяльності, акцентуючи увагу на послідовності дій та відповідних умовах для початку дії. Статус активності відноситься до виконання кожного кроку робочого процесу.

Діаграма діяльності зображена фігурами, з'єднаними стрілками. Стрілки ідуть від початку до кінця дії і представляють послідовний порядок дій. Чорні кола представляють початковий стан робочого процесу. Чорне коло вказує на кінцевий стан. Закруглені прямокутники представляють дії, які описані в тексті всередині кожного прямокутника.

Форма ромба використовується для представлення рішення, що є ключовою концепцією графіка діяльності. Наприкінці перехід (або набір послідовних дій) необхідно вибрати з набору альтернативних переходів для всіх видів використання.

Для представлення паралельних потоків використовуються стрічки синхронізації, які вказують початок або кінець одночасних дій[8].

Діаграма діяльності відображає перехід від зібраної інформації до збереження у середовищі її аналізу на рис.2.2.

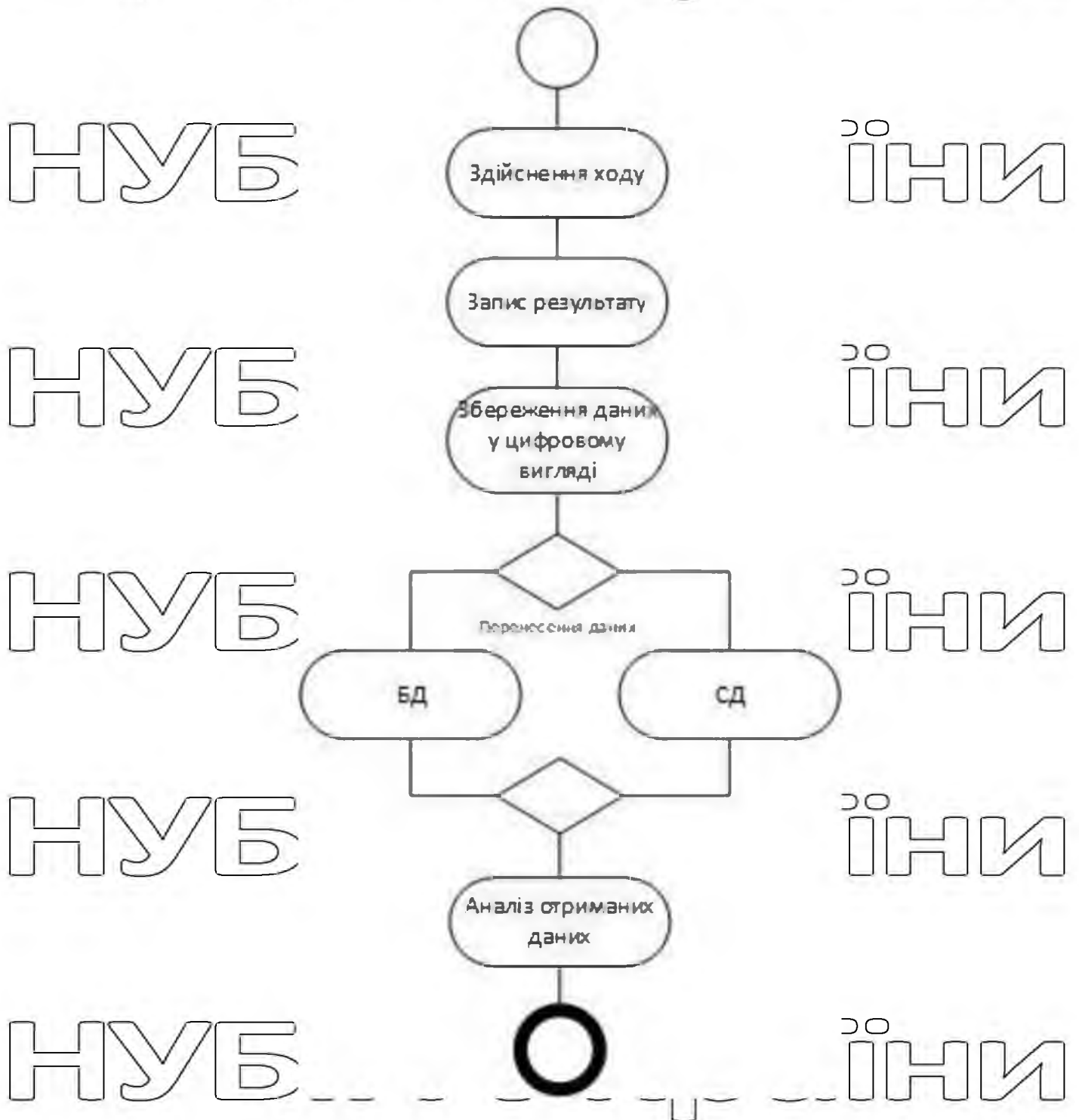


Рис.2.2 Діаграма діяльності

### 2.1.3 Діаграма послідовності

Ця діаграма графічно показує порядок взаємодії певних об'єктів у програмі з часом. Як правило, ця діаграма показує, як користувачі (дійові особи на діаграмі використання) взаємодіють з іншими компонентами

# НУБІП УКРАЇНИ

програми під час реалізації певного використання програми та як взаємодіють інші компоненти програмної системи. Як правило, діаграма послідовності використовується для опису одного із способів використання, зазначених у схемі варіантів використання.

# НУБІП УКРАЇНИ

Діаграми послідовності є способом формалізації випадків використання. Її перевага полягає в тому, що на перших етапах опису сценарію можна визначити склад взаємодіючих компонентів і описати потік повідомлень від одного компонента до іншого. Ці компоненти та потоки повідомлень потім будуть перетворені в конкретні класи (об'єкти), методи цих об'єктів (за термінологією Java). В результаті відразу уточнюється модель системи подій (ДІ), яку ці класи (об'єкти) підтримуватимуть і оброблятимуть [9].

# НУБІП УКРАЇНИ

На діаграмі послідовності показано лише ті об'єкти, які безпосередньо беруть участь у взаємодії. Ця діаграма показує рух дій у часі відповідно від події до реалізації системи рис. 2.3.

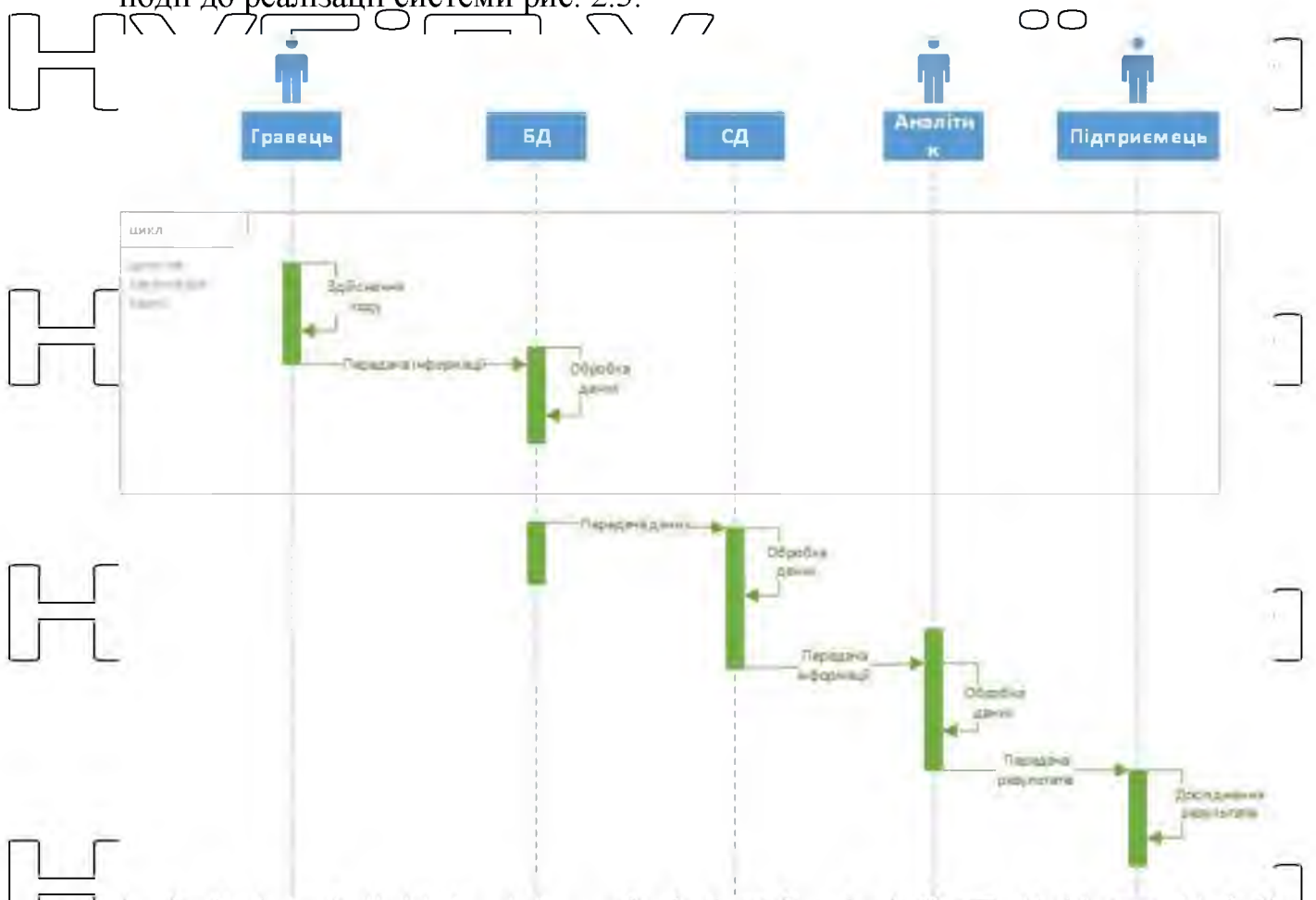


Рис.2.3 Діаграма послідовності

# НУВІП УКРАЇНИ

## З ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Опис архітектури системи і вузлів

Архітектура системи - це набір зв'язків між частинами системи. Існують різні визначення архітектури системи, і різні організації описують її по-різному. Іншими словами, архітектура системи - це сукупність основних функціональних елементів і засобів забезпечення їх взаємодії з одним, з користувачем і з персоналом системи.

Архітектура системи включає:

Структура розгортання системи представлена за допомогою елементів діаграми компонентів.

Метою компонентної діаграми є показати фактичну модель, а також зв'язки та зв'язки між програмними елементами та елементами входу або зовнішніми джерелами даних.

Структура описує наслідування даних рис. 3.1:

1. Зовнішнє джерело. Від джерела до БД – зв'язок один до багатьох.
2. Від БД до СД – зв'язок один до одного.
3. Від СД до модуля аналітики – зв'язок один до одного.
4. Від Модуля аналітики до Аналітика та Підприємця.

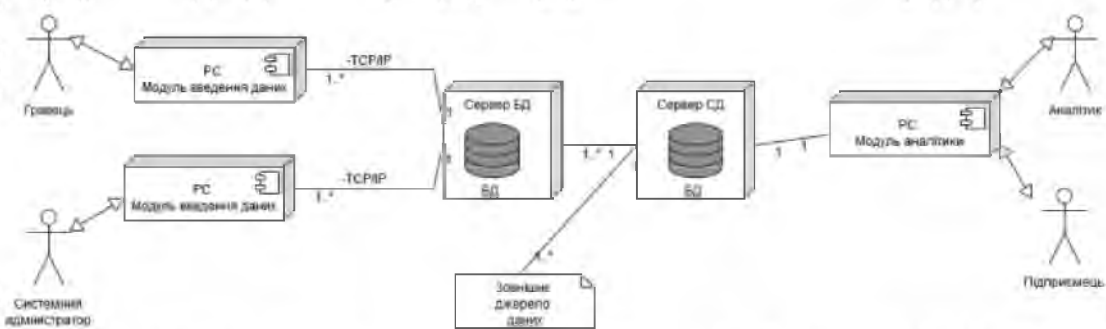


Рис. 3.1. Структура системи

Інформаційно-аналітична система аналізу успішності стратегій складається з гравця, який відповідає за наповнення журналу гри, системний адміністратор – оновлює а додає інформацію про професії, дані два об'єкта відповідають за наповнення Баз даних, після чого частина даних копіюється

# НУВІП УКРАЇНИ

в сховище даних оскільки не вся інформація потрібна аналітику для аналізу поставлених задач. В кінці ми маємо підсистему аналізу або так звану OLAP технологію, за допомогою якої буде відбуватися аналіз даних, що зберігається в сховищі.

### 3.2 Опис вузлів системи, які поставляють дані по сховищу

# НУВІП УКРАЇНИ

Дані до сховища надходять з основної бази даних, що має 6 таблиць. ER діаграма, що створена в додатку ER-win, представлена на рис.3.2



Рис. 3.2 ER діаграма

Розроблена база даних зберігає в собі будь-яку інформацію потрібно для проведення симуляції стратегії, оскільки крім журналу нам потрібне ігрове поле з певними картками, нарахування або списання відповідно та відповідно детальна інформація по професії. Таблиці відображені на рис. 1 цілком задовольняють потреби для проведення симуляцій.

### 3.3 Загальні поняття з напрямку OLAP технології

# НУВІП УКРАЇНИ

Термін OLAP слугує для опису моделі подання даних і відповідно технології їхньої обробки в сховищах даних. В OLAP застосовується багатомірне подання агрегатних даних для забезпечення швидкого доступу до стратегічно важливої інформації з метою поглибленого аналізу. Додатки OLAP повинні мати такі основні властивості:

# НУВІП УКРАЇНИ

# НУВІП УКРАЇНИ

- багатомірне подання даних,  
підтримка складних розрахунків,
- правильний облік фактору часу.

Додатки OLAP опираються на сховища даних, одержуючи від них актуальні дані, що дозволяє контролювати цілісність корпоративних даних.

Таким чином, OLAP - технологія оперативної аналітичної обробки даних, що використовує методи і засоби для збору, зберігання та аналізу багатовимірних даних з метою підтримки процесів прийняття рішень.

Основне призначення OLAP-систем підтримка аналітичної діяльності, довільних запитів користувачів-аналітиків. Мета OLAP-аналізу - перевірка гіпотез.

# НУВІП УКРАЇНИ

Д-р Едгар Ф. Кодд, після свого обширного дослідження реляційної моделі систем баз даних, розробив дванадцять власних правил, які, на його думку, повинні підкорятися базі даних, щоб вважатися справжньою реляційною базою даних.

# НУВІП УКРАЇНИ

Ці правила можуть застосовуватися у будь-якій системі баз даних, яка керує збереженими даними, використовуючи лише свої реляційні можливості. Це базове правило, яке є основою для решти правил.

# НУВІП УКРАЇНИ

## **Правило 1: Інформаційне Правило**

Дані, що зберігаються в базі даних, можуть бути дані користувача або метадані, повинні бути значенням деякої комірки таблиці. Все у базі даних має зберігатися у табличному форматі.

## **Правило 2: Правило гарантованого доступу**

# НУВІП УКРАЇНИ

Кожен окремий елемент даних (значення) гарантовано доступний логічно з комбінацією імені таблиці, первинного ключа (значення рядка) та імені атрибута (значення стовпця). Ніякі інші засоби, такі як покажчики, не можуть бути використані для доступу до даних.

## **Правило 3: Систематичне лікування значень NULL**

# НУВІП УКРАЇНИ



Значення NULL у базі даних мають оброблятися систематично та однаково. Це дуже важливе правило, тому що NULL може інтерпретуватися як одне з таких: дані відсутні, дані невідомі або дані непридатні.

#### **Правило 4: Активний онлайн-каталог**

Опис структури всієї бази даних має зберігатися в онлайн-каталозі, відомому як словник даних, до якого можуть звертатися авторизовані користувачі. Користувачі можуть використовувати ту саму мову запитів для доступу до каталогу, яку вони використовують для доступу до самої бази даних.

#### **Правило 5: Правило всебіжної підмови даних**

Доступ до бази даних можливий лише з використанням мови, що має лінійний синтаксис, який підтримує визначення даних, маніпулювання даними та операції керування транзакціями. Цю мову можна використовувати безпосередньо або за допомогою будь-якої програми. Якщо база даних дозволяє доступу до даних без цієї мови, це вважається порушенням.

#### **Правило 6: Перегляд правила оновлення**

Усі уявлення бази даних, які теоретично можуть бути оновлені, також мають оновлюватись системою.

#### **Правило 7: правило високого рівня для вставки, оновлення та видалення**

База даних повинна підтримувати високорівневу вставку, оновлення та видалення. Це не повинно бути обмежено одним рядком, тобто воно також має підтримувати операції об'єднання, перетину та мінусу для отримання наборів записів даних.

#### **Правило 8: фізична незалежність даних**

Дані, що зберігаються в базі даних, повинні бути незалежними від програм, які звертаються до бази даних. Будь-які зміни у фізичній структурі бази даних не повинні впливати на доступ до даних зовнішніх додатків.

#### **Правило 9: логічна незалежність даних**

Логічні дані у базі даних мають бути незалежними від подання користувача (додачки). Будь-які зміни в логічних даних не повинні впливати

на програми, що їх використовують. Наприклад, якщо дві таблиці об'єднані або одна розбита на дві різні таблиці, це не повинно вплинути або змінити програму користувача. Це одне із найскладніших правил для застосування.

#### **Правило 10: Чесність Незалежності**

База даних має бути незалежною від програми, яка її використовує. Усі обмеження цілісності можуть бути незалежно змінені без будь-яких змін у додатку. Це правило робить базу даних незалежною від інтерфейсної програми та її інтерфейсу.

#### **Правило 11: Незалежність розподілу**

Кінцевий користувач не повинен бачити, що дані розподілені по різних місцях. У користувачів завжди має бути враження, що дані розташовані лише на одному сайті. Це вважається основою розподілених систем баз даних.

#### **Правило 12: Правило без підірвної діяльності**

Якщо в системі є інтерфейс, що забезпечує доступ до записів низького рівня, то інтерфейс не повинен бути здатним підірвати систему та обійти обмеження безпеки та цілісності[10].

Всі дані в сховищі даних поділяються на категорії:

- детальні дані;
- агреговані дані;
- метадані.

Детальні дані - дані, які переносяться безпосередньо від оперативних джерел інформації (OLTP). Вони відповідають елементарним подіям, що фіксуються в звичайних БД. Всі дані поділяються на виміри і факти. Вимірами називаються набори даних, які необхідні для опису подій (студенти, факультети і т.ін.). Вимір є аналогом домену в реляційній моделі. Виміри грають роль індексів для ідентифікації конкретних значень в комірках гіперкуба. Фактами називаються дані, які відображають сутність події

(результати екзамену, кількість студентів і т.п.). Непотрібні детальні дані можуть зберігатися в архівах у стислому вигляді.

Агреговані дані - дані, які отримують агрегуванням детальних даних по певних вимірах. Частина агрегованих даних безпосередньо зберігається в сховищі даних, а не обчислюється при виконанні запитів.

Метадані - це високорівневі засоби відображення інформаційної моделі. Метадані містять таку інформацію: опис структури даних сховища, структури даних, які імпортуються з різних джерел, відомості про періодичність імпортування, методах завантаження і узагальнення даних, засобах доступу і правилах представлення інформації, оцінки витрат часу на отримання відповіді на запит. Метадані знаходяться в репозиторії метаданих [11].

### 3.4 Структура бази та сховища даних

Сховище даних (англ. data warehouse) - предметно орієнтований, інтегрований, незмінний набір даних, що підтримує хронологію і здатний бути комплексним джерелом достовірної інформації для оперативного аналізу та прийняття рішень. В основі концепції сховища даних (СД) лежить розподіл інформації, що використовують в системах оперативної обробки даних (OLTP) і в системах підтримки прийняття рішень (СППР). Такий розподіл дозволяє оптимізувати як структури даних оперативного зберігання для виконання операцій введення, модифікації, знищення та пошуку, так і структури даних, що використовуються для аналізу. В СППР ці два типи даних називаються відповідно оперативними джерелами даних (ОДД) та сховищем даних.

У сховищі даних схема використовується для визначення способу організації системи з усіма сутностями бази даних (таблицями фактів, таблицями розмірів) та їх логічним об'єднанням.

1) Розклад зірск

Це найпростіша та найефективніша схема в сховищі даних. Таблиця фактів у центрі, оточена таблицями з кількома розмірами, нагадує зірку в моделі зіркової схеми.

Таблиця фактів підтримує співвідношення 'один до багатьох' з усіма таблицями розмірностей. Кожен рядок таблиці фактів пов'язаний зі своїми рядками таблиці розмірностей із посиланням на зовнішній ключ.

Через вищевказану причину навігація між таблицями в цій моделі є простою для запитів зведених даних. Кінцевий користувач може легко зрозуміти цю структуру. Отже, всі інструменти бізнес-аналітики (BI) значною мірою підтримують модель схеми Star.

При розробці схем зірок таблиці розмірів цілеспрямовано денормуються. Вони широкі, мають багато атрибутів для зберігання контекстних даних для кращого аналізу та звітності.

**Переваги зіркової схеми**  
Запити використовують дуже прості об'єднання під час отримання даних, завдяки чому продуктивність запитів підвищується.

Отримати дані для звітності просто в будь-який момент часу за будь-який період.

**Недоліки зіркової схеми**  
Якщо у вимогах є багато змін, існуючу схему зірок не рекомендується змінювати та використовувати повторно в довгостроковій перспективі.

Надлишковість даних - це більше, оскільки таблиці не розділені ієрархічно.

**2) Схема Snowflake**  
Зіркова схема діє як вхідні дані для проектування схеми Snowflake.

Снігові відшарування - це процес, який повністю нормалізує всі таблиці розмірностей зі схеми зірок.

Розташування таблиці фактів у центрі, оточене безліччю ієрархій розмірних таблиць, виглядає як Snowflake у моделі схеми Snowflake. Кожен

# НУБІП України

рядок таблиці фактів пов'язаний зі своїми рядками таблиці розмірностей із посиланням на зовнішній ключ.

## 3) Діаграма Галактики

Схема галактики також відома як схема сузір'я фактів. У цій схемі декілька таблиць фактів мають однакові таблиці вимірів. Розташування таблиць фактів та таблиць розмірностей виглядає як сукупність зірок у моделі схеми Галактики.

Спільні розміри в цій моделі відомі як відповідні розміри.

Цей тип схеми використовується для складних вимог та для сукупних таблиць фактів, які є більш складними для підтримки схеми Star (або) схеми Snowflake. Цю схему важко підтримувати через її складність.

Під час проектування схем Snowflake таблиці розмірів цілеспрямовано нормалізуються. Зовнішні ключі будуть додані до кожного рівня таблиць розмірів для посилання на його батьківський атрибут. Складність схеми Snowflake прямо пропорційна рівням ієрархії таблиць розмірностей.

## 4) Схема скупчення зірок

Схема Snowflake з багатьма таблицями розмірностей може потребувати більш складних об'єднань під час запиту. Зіркова схема з меншою кількістю таблиць розмірів може мати більшу надмірність. Отже, схема зіркового скупчення з'явилася на зображенні, поєднавши ознаки двох наведених вище схем.

Зіркова схема є основою для розробки схеми зоряного кластера, і кілька основних таблиць розмірів із зіркової схеми оброблені сніжинками, і це, в свою чергу, утворює більш стабільну структуру схеми [12].

Для зберігання даних ми використали структурну схему «зірка», де маємо 3 виміри та одну таблицю фактів.

Структура сховища даних представлена на рис. 3.3.

# НУБІП України

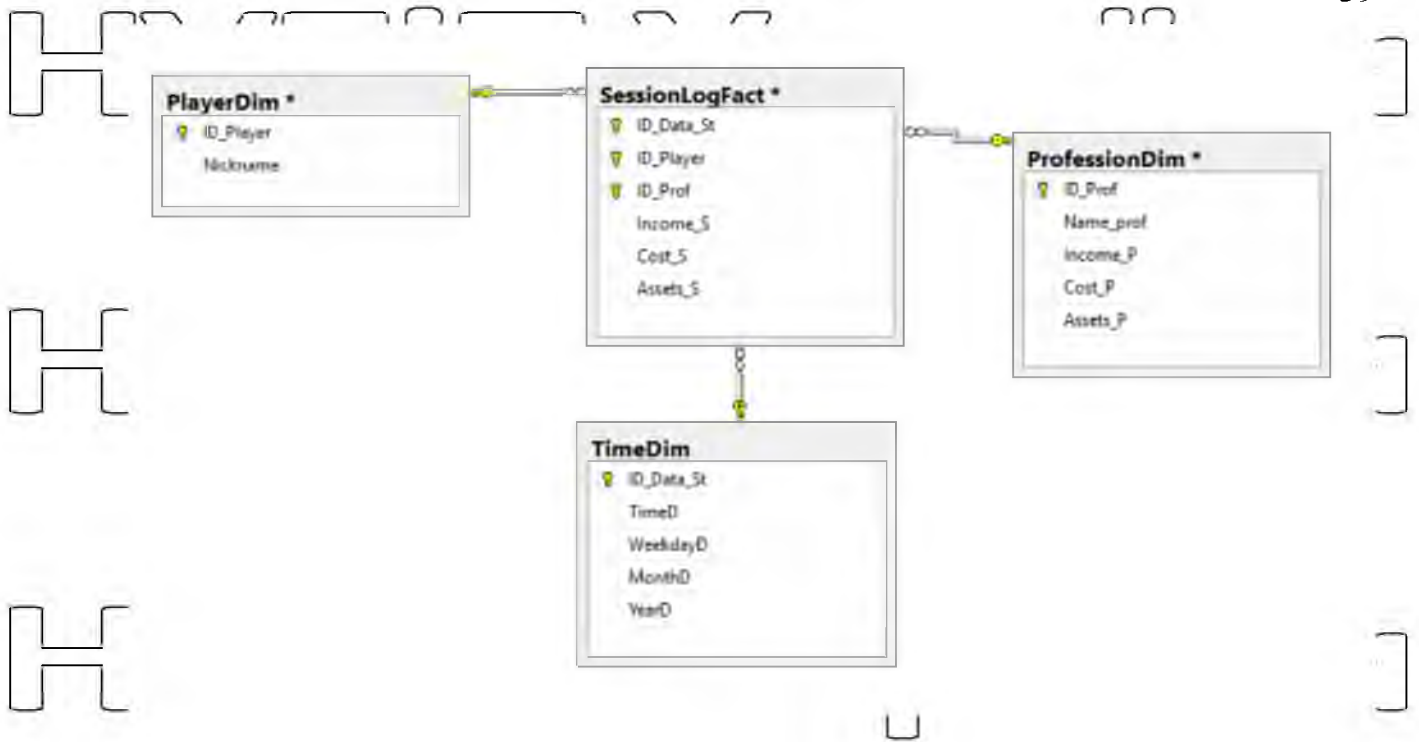


Рис. 3.3 Структура сховища даних

На даному етапі роботи покажемо особливості заповнення сховища даних. Для заповнення вимірів та таблиці фактів було написано запити, що вставляли та зберігали інформацію. Запити для заповнення таблиць див. в додатку А.

### 3.5 Збір інформації та її впровадження

#### Методи збору інформації

Цей блок фокусується на отриманні інформації із зовнішніх джерел і реалізує сховище, яке обговорюється в наступному блоці. Особа та автоматизований модуль можуть відповідати за збір інформації. Джерело – файли excel та word, надані працівниками деканату, кафедри та інформація про сайт

#### Способи зберігання інформації

Другий крок використовує перший блок і зберігає інформацію, з якою ви можете працювати, тобто мати справу. Носієм є база даних/сховище даних (система OLAP)

# НУБІП України

## Методи обробки інформації

Третій блок є модулем для організації інформації для її підготовки до аналізу і надалі для різних інформаційних потреб.

## Режими відображення інформації

Четвертий блок стосується методів створення та побудови логіки роботи з картографією та звітами, при необхідності - модулів інтерфейсу, аналізу й обробки.

## Опис таблиць що належать до сховища даних

PlayerDim – містить інформацію про гравців, оскільки в різних сеансах симуляції можуть приймати участь одні і ті ж гравці:

ProfessionalDim – містить інформацію про професію, оскільки в різних сеансах симуляції можуть приймати участь одні і ті ж професії:

TimeDim – фіксує час подій;

SesionLogFact - до таблиці фактів мігрують всі ключі з таблиць вимірів, окрім цього ми зберігаємо інформацію про нарахування та списання коштів та про грошовий потік.

НУБІП України

НУБІП України

НУБІП України

# 4 РОЗРОБКА СИСТЕМИ АНАЛІЗУ ДАНИХ

## 4.1 Механізм вилучення, обробки і передачі даних

### 4.1.1 Опис BI та створення в його середовищі проекту служби SSAS

BI не замінює, а розвиває та доповнює традиційні інструменти аналізу, відомі до 1990-х років. BI також не замінює висококваліфікованих і досвідчених аналітиків, а доповнює та змінює їхні здібності, їхні знання, покращує їх аналітичне мислення – тому що впроваджено на основі ідеалізованих алгоритмів, типових для мислячих аналітиків. Інструменти BI дозволяють швидко отримувати потенційно корисні нетривіальні знання з первинних даних та їх візуалізацію для прийняття більш корисних рішень, які без них недоступні аналітичним робочим групам будь-якого розміру, інтелектуальної сили та досвіду.

Синергія в цьому контексті означає, що автономні компоненти комплексу можуть використовуватися окремо або разом, у різних ієрархічних підпорядкуваннях, а їх комбіноване застосування дає новий ефект, який виходить за рамки простої суми окремих компонентів. Ієрархія в цьому контексті означає, що часткові інструменти повинні бути розроблені постачальниками та впроваджені клієнтами, беручи до уваги необхідність об'єднання їх для аналізу – на рівні програмного забезпечення та інформації. Кожен підхід має суттєві недоліки: синергія нестабільна і несумісна, а ієрархія організована й консервативна.

BI зазвичай поєднує наступні окремі технології:

- технології реляційних баз даних (на даний момент нереляційні бази даних - NoSQL)
- найзріліші технології штучного інтелекту,
- передові технології традиційної статистики,
- передові технології перегляду результатів аналізу.

Business Intelligence у вузькому розумінні поєднує в собі наступні автономні та напівавтономні інструменти:



# НУБІП України

інструменти ETL (Extract, Transform, Load) - інструменти для вилучення, перетворення та завантаження інформації як у Business Intelligence, так і між підсистемами,

- спеціальні статистичні інструменти,

# НУБІП України

ресурси для нерегульованих запитів (ad hoc запити),  
інструменти швидкого багатовимірного аналізу OLAP,

- специфічні засоби Data Mining та Visual Mining «викопування

даних»,

- спеціальні інструменти візуалізації (інформаційні панелі, система

# НУБІП України

показників),

- елементи експертних систем,

- специфічні засоби "текстовидобутку" Text Mining в даний час,

- інші.

# НУБІП України

Найкорисніші результати інструменти BI дають результати аналізу інформації в словищах даних, але можуть принести корисний ефект у витринах даних (Data Marts), навіть на 100 наборах даних або менше.

# НУБІП України

Найефективніші інструменти Business Intelligence (елементи OLAP, Data Mining, Visual Mining, Video Mining, Web Mining і Text Mining і Opinion Mining) вимагають потужного апаратного та програмного забезпечення, деякі з цих інструментів були доступні з 1990-х років, а інші з другого десятиліття 20-го століття[13].

# НУБІП України

При виконанні магістерського дослідження було використано SQL Server Business Intelligence, в якому формувався куб, створювались потоки даних для отримання даних з джерел на заповнення створеного кубу.

Для побудови кубу було створено проєкт (створення рішення) типу

Analysis Service (рис.4.1).

# НУБІП України



Рис. 4.1 Створення проекту

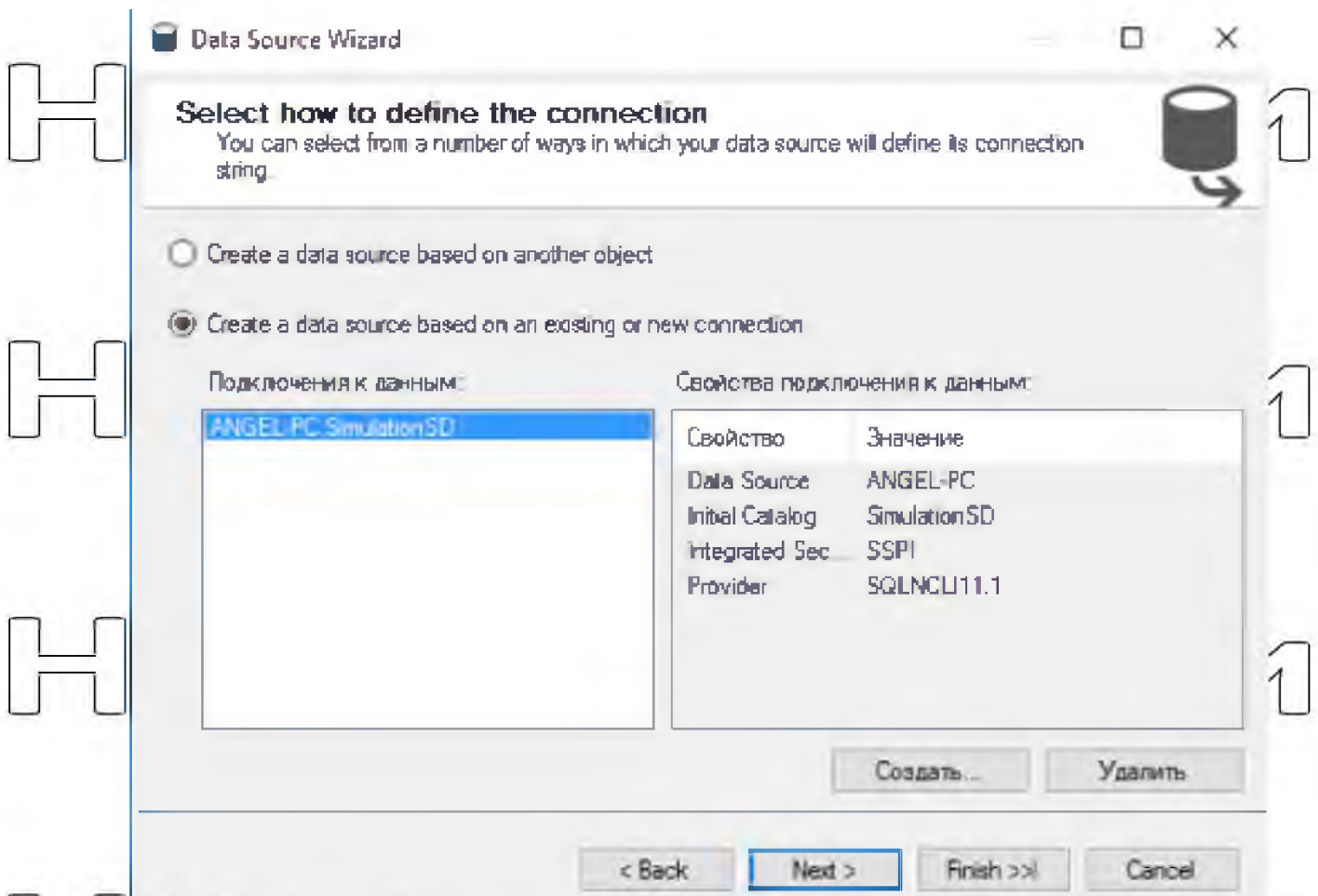


Рис. 4.2 Створення підключення до джерела даних

НУБІП України

Наступним кроком є створення Data Source View (рис. 4.3). Під Data Source View розуміється зріз джерела, яке буде використовуватися для заповнення сховища, при цьому в нього можуть входити як таблиці, так і уявлення (view) реляційної бази - джерела даних.

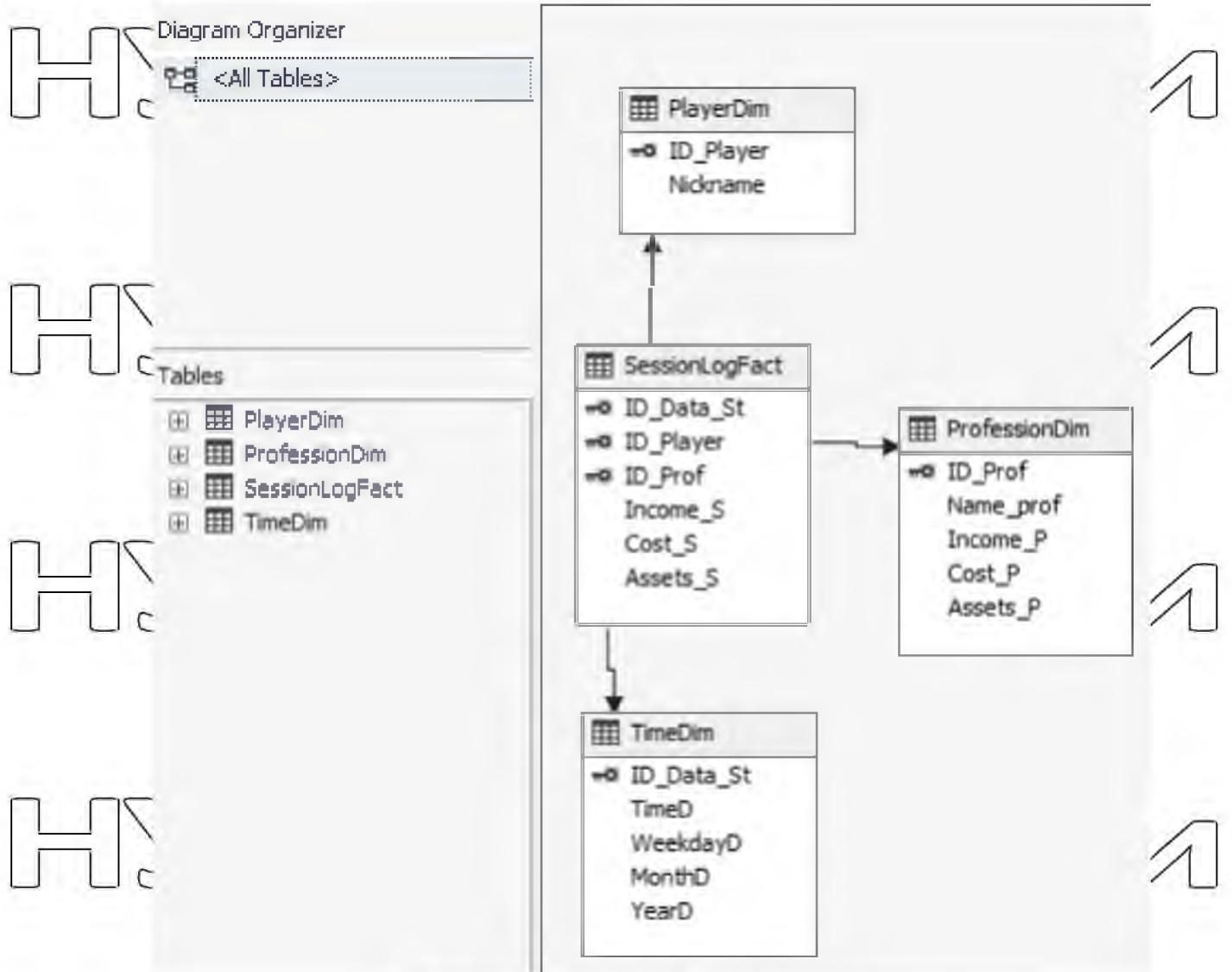


Рис. 4.3 Результат Data Source View

НУБІП України

Наступним кроком є власне розгортання кубу, де на рис. 4.4 відображено повномасштабну схему кубу.

НУБІП України

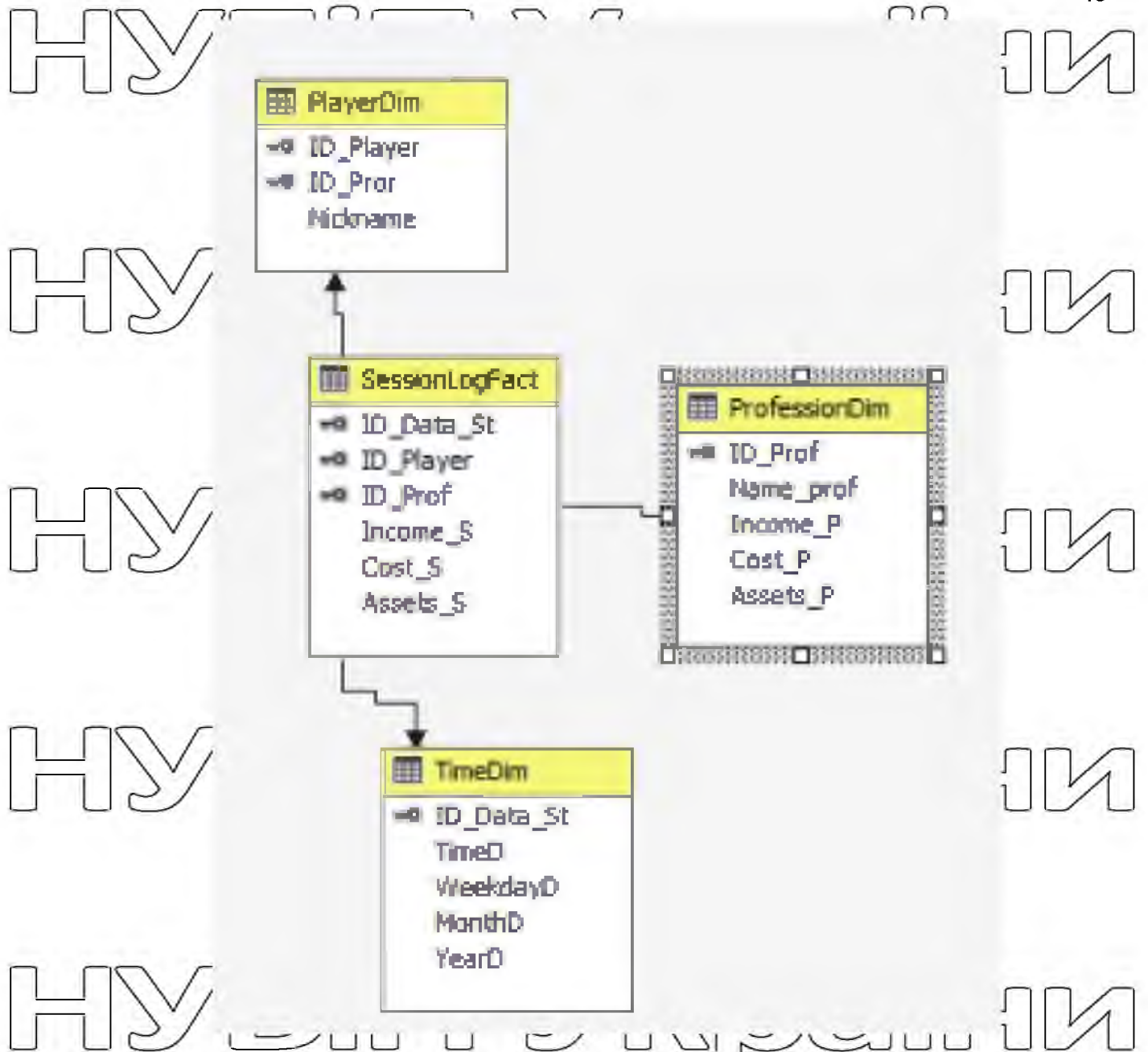


Рис. 4.4 Розгорнутий куб

#### 4.1.2 Реалізація отримання даних за допомогою Data Flow

Завданням «Потоку даних» є інкапсуляція механізму потоку даних, який переміщує дані між джерелами та місцями призначення, і дозволяє користувачеві перетворювати, очищати та змінювати дані під час їх переміщення. Додавання завдання потоку даних до потоку керування пакетом дає змогу пакету витягувати, перетворювати та завантажувати дані.

Потік даних складається щонайменше з одного компонента потоку даних, але зазвичай це набір пов'язаних компонентів потоку даних: джерела,

які витягують дані; перетворення, які змінюють, маршрутизують або узагальнюють дані; і пункти призначення, які завантажують дані.

Під час виконання завдання потоку даних створює план виконання з потоку даних, а механізм потоку даних виконує план. Ми можемо створити завдання потоку даних, яке не має потоку даних, але воно виконується, лише якщо воно включає принаймні один потік даних.

Щоб масово вставити дані з текстових файлів у базу даних SQL Server, можна використовувати завдання масової вставки замість завдання потоку даних і потоку даних. Однак завдання масової вставки не може перетворити дані.

Завдання потоку даних може містити кілька потоків даних. Якщо завдання копіює кілька наборів даних, і якщо порядок, в якому дані копіюються, не є значним, може бути зручніше включити кілька потоків даних до завдання Потоку даних. Наприклад, можна створити п'ять потоків даних, кожен з яких копіює дані з плоского файлу в іншу таблицю вимірів у схемі зірки сховища даних.

Однак, обробник потоку даних визначає порядок виконання, коли є кілька потоків даних в межах одного завдання потоку даних. Тому, коли порядок важливий, пакет повинен використовувати кілька завдань потоку даних, кожне завдання, що містить один потік даних. Потім можна застосувати обмеження пріоритету для керування порядком виконання завдань [14].

Отримання даних з джерела та заповнення згенерованого кубу було виконане за допомогою Data Flow. Для цього потрібно створити проект служб SSIS (рис. 4.5).

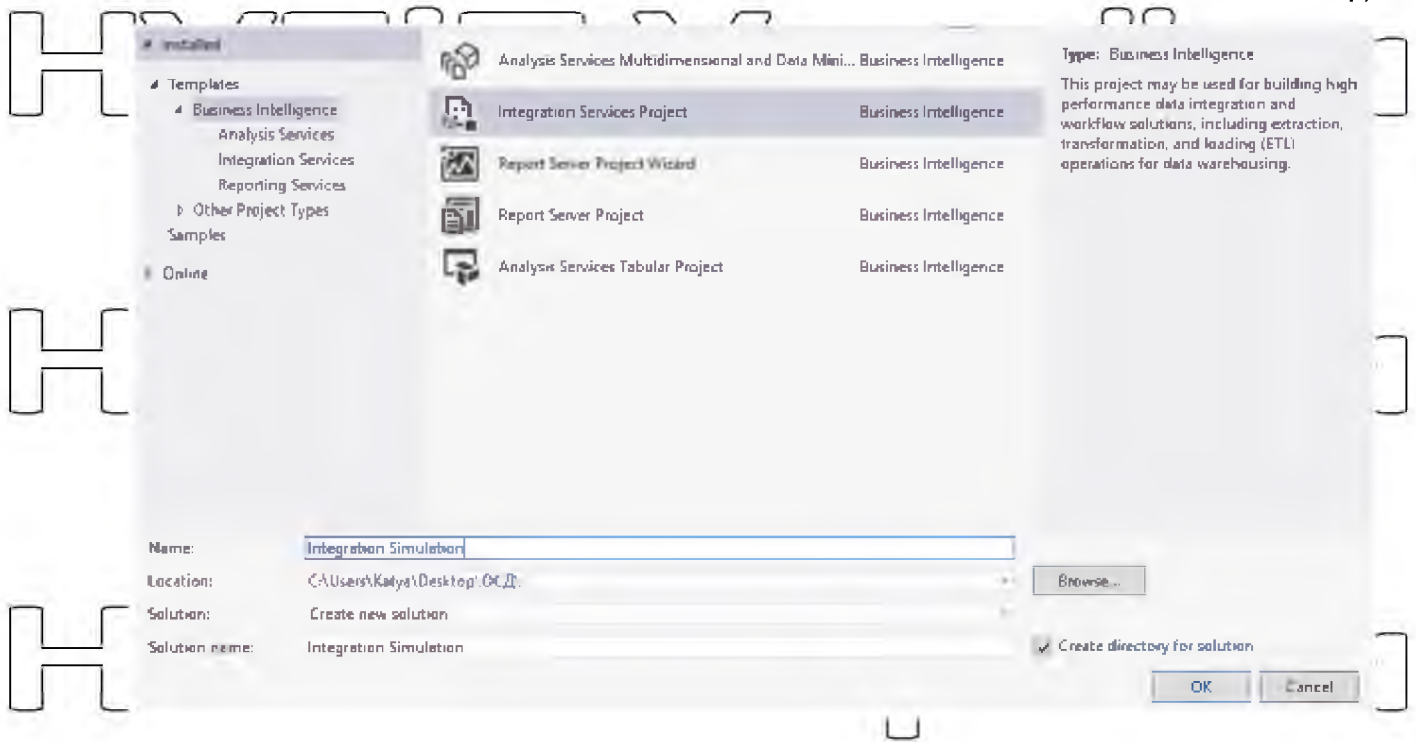


Рис. 4.5 Створення проекту для заповнення кубу даними

Потім створити задачу потоку даних для вимірів першої черги (рис. 4.6), а в ній створити потоки ланих для кожного виміру (рис.4.7) та факту (рис.4.8).

На скріншотах представлено запущений проєкт, де видно позитивний результат виконання.



Рис. 4.6 Задача потоку управління даними

НУБІП України



Рис. 4.7 Потік вимірів

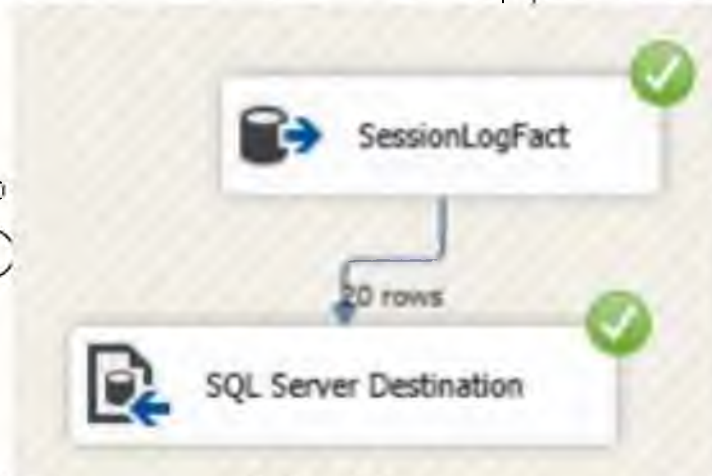


Рис. 4.8 Потік фактів

# 5 ВПРОВАДЖЕННЯ СИСТЕМИ АНАЛІЗУ ДАНИХ

## 5.1 Основні відомості про аналіз даних у середовищах.

Для створення звітів використовується програмне забезпечення Power

BI.

Основні переваги цієї програми:

- Перетворюйте дані на приголомшливі зображення та діліться ними з колегами в будь-якому пристрої;
- Візуально вивчайте та аналізуйте дані в таблицях, базах даних, сховищах або в хмарі — все в одній програмі;
- Співпрацювати та обмінюватися індивідуальною інформацією за допомогою інтерактивних інформаційних панелей та звітів;
- Покращуйте свою систему за допомогою інтегрованого управління та захисту.

При створенні звітів враховується точність і точність вихідних даних, з якими аналітик працюватиме в майбутньому. Відповідно до розділу 1.1 щодо визначення завдання надано перелік питань аналізу даних, який відображається у службі звітності.

Служба звітності — це технологія для створення звітів у середовищі Power BI. Звіти створюються шляхом збереження графічних зображень у файлі PDF. Data Analysis Visual Studio працює через сховище даних і зв'язок між метриками та вибраними фактами та показниками, щоб обробити інформацію та подати її до служби аналізу та, якщо необхідно, до служби звітності.

Обидві технології працюють із вбудованим гіперкубом і можуть використовувати метрики для аналізу даних із сховища та створення звітів на основі цих даних. Разом це дозволяє аналізувати дані з різних сторін.



# НУБІП України

## 5.2 Звіт в середовищі Power BI

Провівши сеанси симуляції було отримано набір даних, які були передані на аналіз для дослідження.

Дослідження проводилися по трьох професіям та трьох стратегіям.

Перша професія та стратегія полягає в тому, що гравець, який проходить симуляцію не ризикує своїми фінансами а старатється плавно збільшувати свої пасивні доходи та активні фінанси. На рис. 5.1 було створено діаграму з листом прогнозу, де ми можемо побачити прогноз на інші ходи (ітерації), які ще не були здійснені.



Рис. 5.1 Прогноз заробітньої плати викладача

По даному графіку можемо сказати, що навіть в найгірших випадках будемо мати стабільне підвищення заробітної плати, що не може не значити, що стратегія гравця, який відповідає за дану професію має сенс і право на життя. Результати даного прогнозу ми можемо побачити на рис. 5.2.

Номер ходу	Заробітна плата	Прогноз по заробітній платі	Прив'язка малої ймовірності (до заробітній платі)	Прив'язка високої ймовірності (до заробітній платі)
542	142270	142270		142270
543		145077,5524	138050,2066	148104,4982
544		148284,7048	138201,62	149467,7999
545		148492,0572	138156,9485	148837,7159
546		148699,4099	138214,2948	149184,3604
547		148906,7627	138274,545	149538,579
548		149114,1154	138338,2422	149889,9855
549		149321,4681	138403,9547	150238,9789
550		149528,8208	138471,9117	150585,7967
551		149736,1736	138542,9527	150930,3004
552		149943,5264	138613,9402	151273,1838
553		150150,8794	138687,7478	151614,005
554		150358,2322	138763,2911	151953,1465
555		150565,5852	138840,5781	152290,7892

Рис. 5.2 Таблиця з розрахунками можливих показників

Друга професія та стратегія полягає в тому, що гравець, який проходить симуляцію мінімально починає ризикувати своїми фінансами, тобто для професії хірурга гравець вибрав певний алгоритм, а саме:

- При будь-якій можливості збільшити пасивний дохід, вкладати в

це кошти.

- При пропозиції придбати акції гравець повинен подумати, яку кількість акцій придбати, але головною умовою є те, що гравець може придбати акції не більше ніж на половину своїх активних фінансів.

- По можливості купувати будинки, землю і тд. Оскільки нерухомість в подальшому є можливість продати і підвищити активні фінанси

На рис. 5.3 було створено діаграму з листом прогнозу, де ми можемо побачити прогноз на інші ходи (ітерації), які ще не були здійснені.



Рис. 5.3 Прогноз заробітньої платі хірурга

По даному графіку можемо сказати, що в найгірших випадках будемо мати зменшення заробітної плати, що не може не значити, що стратегія даного гравця, який відповідає за дану професію не докінця продумана оскільки навіть в найгіршому випадку ми повині мати, як мінімум стабільне значення яке не буде зменшуватися. Результати даного прогнозу ми можемо побачити

на рис. 5.4.

Номер акції	Заробітна плата	Привязка до заробітної плати	Привязка низької ймовірності (по заробітній платі)	Привязка високої ймовірності (по заробітній платі)
542	135200	135200	135200	135200
543		135357,3794	130967,791	139746,9659
544		135514,7589	130605,0874	140424,4303
545		135672,1383	130290,6125	141053,6642
546		135829,5178	130012,6715	141646,2641
547		135986,8972	129763,513	142210,2565
548		136144,2767	129537,9654	142750,648
549		136301,6561	129331,8955	143271,4168
550		136459,0356	129142,5813	143775,4899
551		136616,415	128967,6805	144263,1405
552		136773,7945	128805,4135	144742,1755
553		136931,1739	128654,289	145208,0589

Рис. 5.4 Таблиця з розрахунками можливих показників

Третя професія та стратегія полягає в тому, що гравець, який проходить симуляцію максимально ризикує своїми фінансами, тобто для професії помічника депутата гравець вибрав для себе певний алгоритм, а саме:

- При будь-якій можливості збільшити пасивний дохід, вкладати в це кошти.
  - При пропозиції придбати акції, гравець 100% вкладає всі наявні фінанси;
  - По можливості купувати будинки, землю і тд. Оскільки нерухомість в подальшому є можливість продати і підвищити активні фінанси.
- На рис. 5.5 було створено діаграму з листом прогнозу, де ми можемо побачити прогноз на інші ходи (ітерації), які ще не були здійснені.



Рис. 5.5 Прогноз заробітної плати помічника депутата

По даному графіку можемо сказати, що в найгірших випадках будемо мати зменшення заробітної плати, що не може не значити, що стратегія даного

гравця, який відповідає за дану професію не докінця продумана оскільки навіть в найгіршому випадку ми повині мати, як мінімум стабільне значення

яке не буде зменшуватися, але приймаючи до уваги те, що гравець вкладає на різних етапах всі свої кошти, то можемо сказати, що результати перевищили

всі заплановані показники, оскільки такою стратегією гравець може вийти з симуляції завчасно, так як високі ризики можуть залишити гравця без фінансів. Результати даного прогнозу ми можемо побачити на рис. 5.6.

Номер ходу	Заробітна плата	Прогнозована заробітна плата	Прогнозована вартість ймовірності (по заробітній платі)	Прогнозована вартість ймовірності (по заробітній платі)
542	440600	440600	440600	440600
543		441190,2724	429045,3199	453245,1448
544		441790,4647	429720,8411	466880,0894
545		442385,6971	430401,4724	480569,9218
546		442980,1294	431093,6897	494276,1699
547		443576,1618	431800,7001	508049,6245
548		444171,3942	432523,8933	521785,095
549		444766,6265	433271,5927	535496,6403
550		445361,8589	434037,3502	549187,3679
551		445957,0912	434821,6301	562860,3884
552		446552,3235	435623,9494	576518,7928
553		447147,5559	436443,8227	590165,6899
554		447742,7882	437281,7003	603804,1899
555		448338,0207	438137,8216	617428,3928
556		448933,253	439012,6777	631042,4994

Рис. 5.6 Таблиця з розрахунками можливих показників

Далі дослідимо показники опираючись одночасно на всі три стратегії.

Оскільки в симуляції ми передбачали можливість кредитів для придбання нерухомості був витрачений час на підбір та дослідження по якому принципу обраховувати відсотки і як взагалі проводити обрахунки і чи зазначати максимальну суму по кредитувані.

Головним критерієм вибору вигідного кредиту є відсоткова ставка, яка є прямим показником його вартості для одержувача. Саме за цим критерієм

банки змагаються один з одним, навпередбій пропонуючи низьку вартість, перші місяці без оплати і навіть нульову ставку. Але далеко не завжди заявлені

умови відповідають дійсності, і при глибшому вивченні клієнту дізнається, що кінцева вартість послуги буде набагато вищою за зазначену.

Процентна ставка означає збільшення кінцевої грошової суми, яку споживач повинен буде повернути кредитору після закінчення терміну дії

договору. У банківських кредитах зазначається річна вартість, яка на споживчі позики залишається однією з найвищих, утримуючись на рівні 44% річних.

Кількість та розмір щомісячних платежів ділиться на весь період користування

таким чином, щоб позичальник не мав змоги зменшити кінцеву вартість кредиту. У деяких випадках це прописує безпосередньо в договорі, в інших при достроковому погашенні клієнт потрібно буде покрити збитки банку, виплативши комісію.

При оформленні позики в банку важливо розуміти, що зазначений у договорі %, навідше за все, номінальна, тоді як реальна кінцева вартість послуги дещо вища. Саме розрахунок реальної ціни кредиту вкрай важливий у виборі вигідної пропозиції, щоби не довелося переплачувати банку, і щоб кінцева сума не стала неприємним сюрпризом.

При необхідності взяти споживчу позику не важливо, на які цілі, варто враховувати, що її вартість за рік у середньому збільшиться більш ніж на 50%. Сюди закладається і зростання інфляції, і зміни курсу національної валюти, які значно впливають на ціну позики. Щоб правильно порахувати відсотки за кредитом, важливо враховувати:

- суму кредиту;
- величину річного відсотка;
- кількість днів користування;
- тип погашення позики: диференційований чи анuitетний.

Якщо з усіма пунктами списку зрозуміло, останній варто пояснити докладніше. Диференційована система платежів передбачає поступове зменшення суми за рахунок внесення місячних платежів, що покривають і відсоток, і тіло позики. Тобто останній платіж у банк буде набагато меншим, ніж перший. Анuitетна система відрізняється однаковим розміром всіх платежів: від першого до останнього. З огляду на це можна зрозуміти, що принцип розрахунку відсотків у кожному випадку буде дещо відрізнятися [15].

За основу було створено формулу в Excel яка автоматично після внесення суми в кредит а саме якщо внесли для прикладу відображено на рис.

5.7 формула з Excel, а на рис 5.8 результати відповідно до прикладу, де сума кредиту дорівнює 70000.

НУБІП  $=B2+\$B\$2*0,01$  аїни

Рис. 5.7. Формула Excel для обрахунку кредиту

	А	В	С
1			
2		70000	
3		70700	
4		71400	
5		72100	
6		72800	
7		73500	
8		74200	
9		74900	
10		75600	
11			

Рис. 5.8 Результати обрахунків

Оскільки це економічний симулятор то відсотки нараховується не щомісячні, а для кожного ходу. Оскільки деякі покупки були такими що гравці не могли придбати 100% їх за свої кошти, то вони використовували можливості використання кредитів. На рис. 5.9 відображено порівня історії кредитів для трьох різних стратегій.

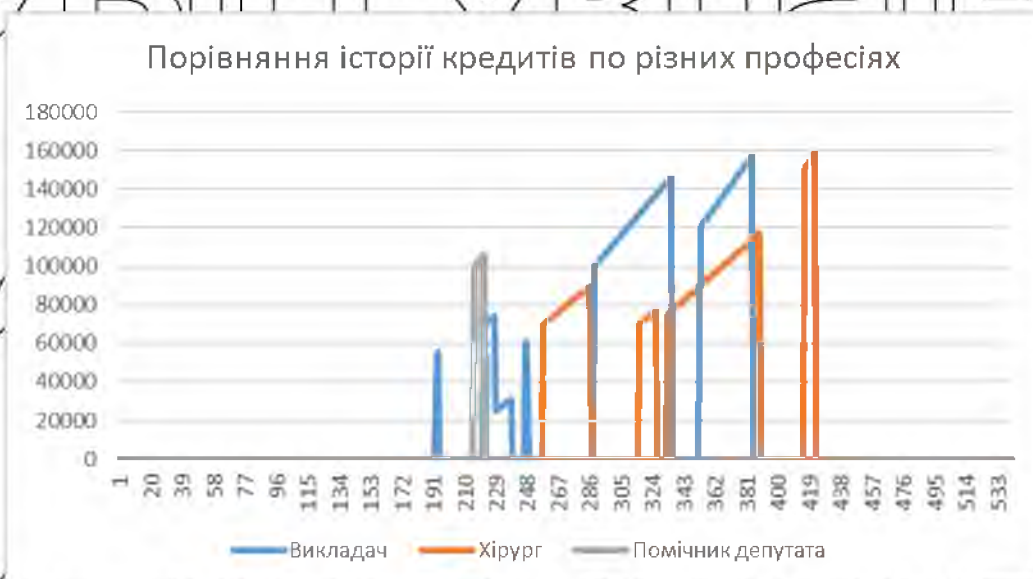


Рис. 5.9 Порівняння історії кредитів

Досліджуючи наявні результати можемо побачити, що викладач в порівнянні з іншими професіями частіше використовував

кредитування для придбання нерухомості, оскільки тільки на нерухомість розповсюджується можливість кредитування. На другому місці по

кредитуванні є гравець який вів професію хірурга, та найменше кредитів брав гравець який відповідав за професію помічника депутата при чому, термін кредитування у нього був найменшим при тому що сума була не маленька.

Далі проведемо порівняння підвищення заробітної плати не в розрізі прогнозування, а в розрізі порівняння між різними професіями рис. 5.10.

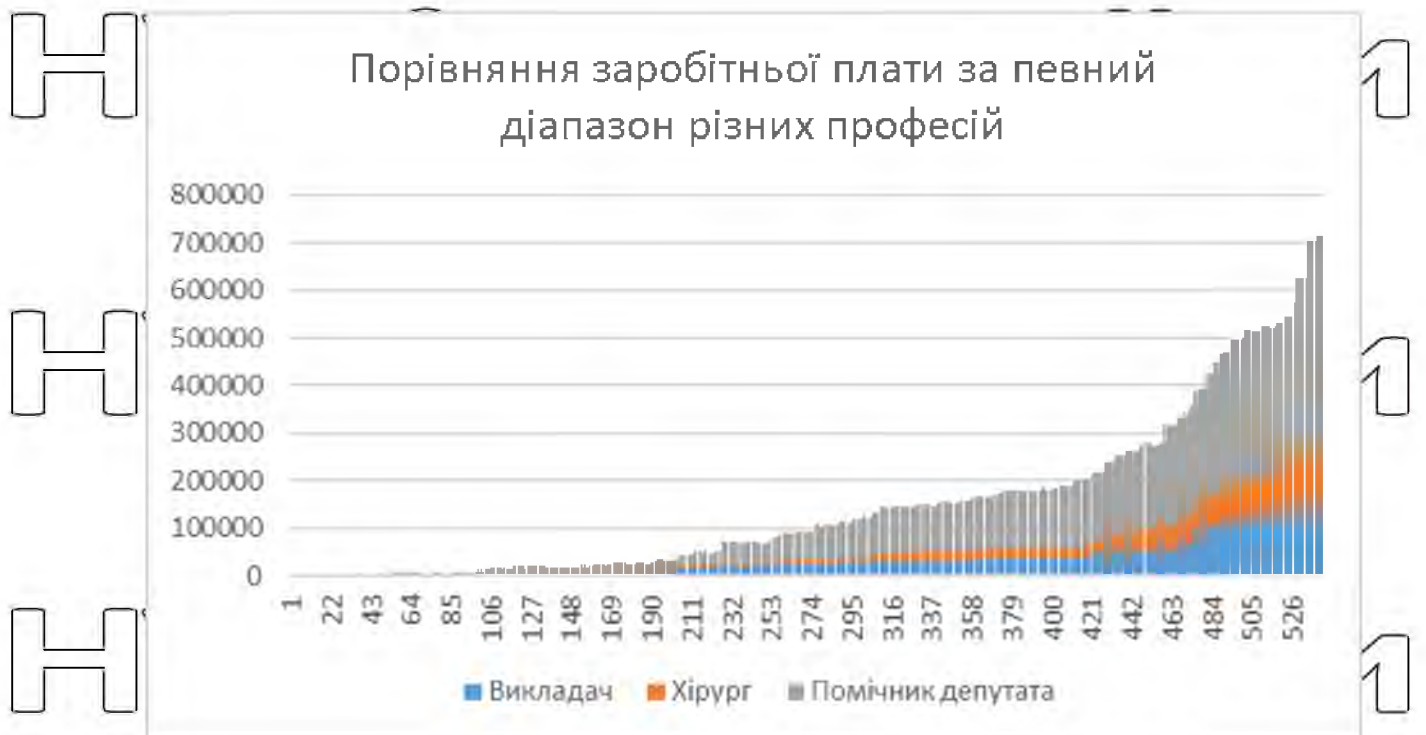


Рис. 5.10 Порівняння заробітної плати різних професій

Оскільки гравець, що відповідав за викладача не ризикував, то він не зміг отримати високої зарплати в порівнянні з іншими гравцями, але порівнюючи з його початковою заробітною платою можна сказати, що гравець досяг доволі високих показників.

Аналізуючи наступну професію Хірурга можемо побачити що доволі довгий час його заробітня плата була майже на рівні з заробітною платою викладача, але оскільки даний гравець уже мін іноді ризикувати він зміг отримати різке підвищення своїх фінансів.

НУБІП України

Досліджуючи третю професію можемо побачити, що використовуючи стратегію максимальних ризиків гравець уже після 232 ходу має значно більше значення заробітної плати чим інші гравці, що відображає що іноді ризик може бути обгрунтованим.

НУБІП України

Але якщо порівнювати витрати рис. 5.11 то можемо побачити, що чим більша заробітна плата то тим більші витрати відповідно до до заробітної плати.



Рис. 5.11 Порівняння витрат по різних професіям

НУБІП України

І на кінець було проведено аналіз активних фінансів паралельно всіх професій рис. 5.12

НУБІП України

НУБІП України



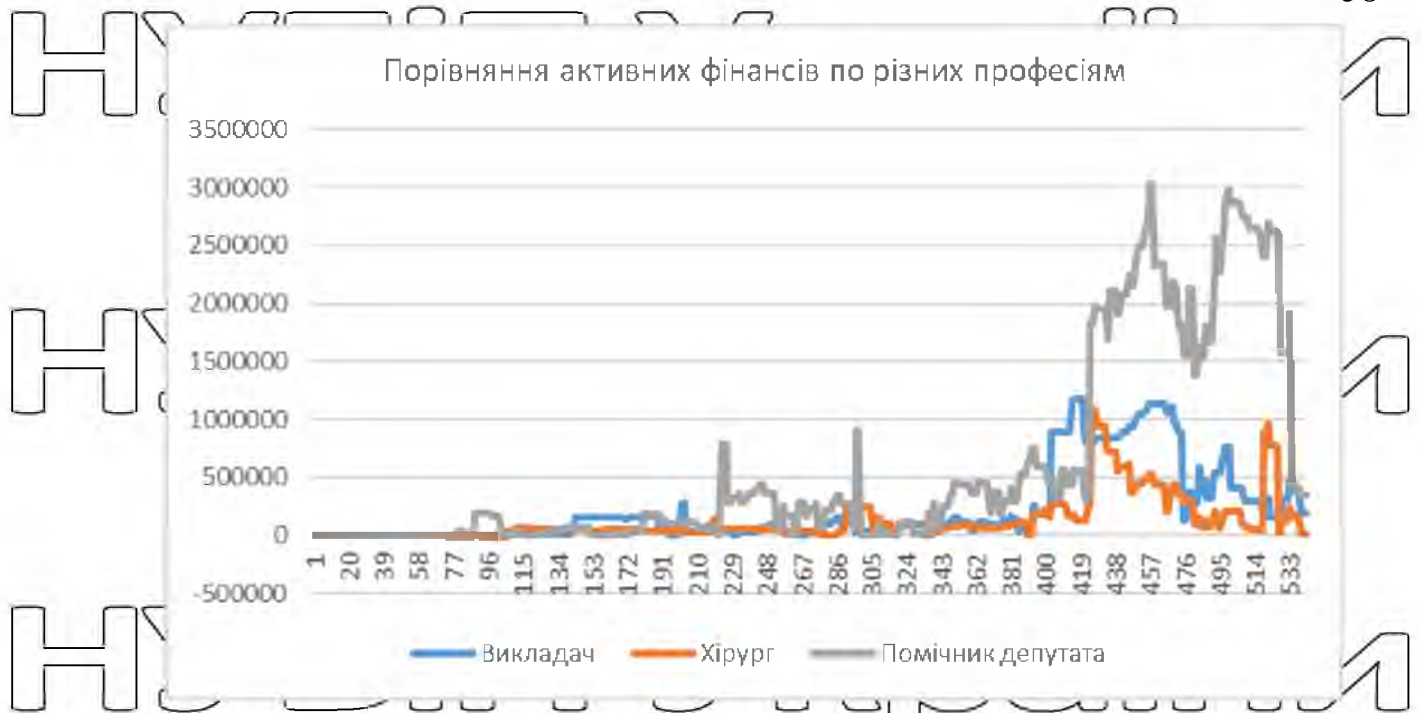


Рис. 5.13 Порівняння активних фінансів

По даним діаграми бачимо, що:

- Викладач, який не ризикував своїми фінансами більш менш мав стабільше збільшення фінансів, при тому, що іноді мав різкі скачки суми в більшу сторону, а іноді навпаки різкі падіння вниз.
- Хірург, який частково ризикував, має схожі показники з викладачем, але оскільки він мав фінанси для додаткових ризиків то мав частіші скачки ніж викладач.
- Помічник депутата, стратегія якого полягала в постійному ризику, гарно дав наочно побачити що великі вклади не завжди були доречні, оскільки після вкладень всіх фінансів не завжди одразу він міг отримати потрібні значення.

### 5.3 Розрахунок KPI

У бізнес-термінології ключовий показник ефективності (KPI) є кількісним виміром для визначення бізнес-цілей. KPI часто оцінюється з часом. Наприклад, в відділ продажів організації може використовувати KPI для вимірювання щомісячного валового прибутку з прогнозованим валовим

прибутком. Для оцінки витрат бухгалтерія може вимірювати місячні витрати та доходи, а відділ кадрів може вимірювати щоквартальну плинність кадрів.

Кожен з них є прикладом KPI. Бізнес-фахівці часто використовують ключові показники ефективності, які згруповані в таблицю показників бізнесу, щоб отримати швидкий і точний історичний підсумок успіху бізнесу або визначити тенденції.

**KPI в табличній моделі включає:**

### 1. Базове значення

Базове значення визначається мірою, яка перетворюється на значення.

Це значення, наприклад, може бути сукупністю фактичних продажів або розрахованим показником, таким як прибуток за певний період.

### 2. Цільове значення

Цільове значення визначається мірою, яка перетворюється на значення, або абсолютним значенням. Наприклад, цільовим значенням може бути сума, на яку бізнес-менеджери організації хочуть збільшити продажі або прибуток.

### 3. Пороги статусу

Поріг статусу визначається діапазоном між низьким і високим порогом або фіксованим значенням. Поріг статусу відображається з графікою, щоб допомогти користувачам легко визначити статус базового значення порівняно з цільовим значенням.

### Створення та редагування KPI

Для створення ключових показників ефективності в конструкторі моделей ви будете використовувати діалогове вікно «Ключовий показник ефективності». Оскільки ключові показники ефективності мають бути пов'язані з мірою, ви створюєте KPI, розширюючи міру, яка оцінюється до базового значення, а потім створюється міра, яка оцінюється до цільового значення, або вводячи абсолютне значення. Після визначення базової міри (значення) і цільового значення ви можете визначити порогові параметри стану між базовими і цільовими значеннями. Статус відображається у графічному форматі за допомогою вибраних піктограм, смужок, графіків або

кільгорів. Базові та цільові значення, а також статус можна потім додати до звіту або зведеної таблиці як значення, які можна порівняти з іншими полями даних.

Щоб переглянути діалогове вікно «Ключовий показник ефективності», у сітці вимірювань для таблиці клацніть правою кнопкою миші мірку, яка буде використовуватися як базове значення, а потім натисніть «Створити KPI». Після того, як показник буде розширено до KPI як базового значення, поруч із назвою міри в сітці показників з'явиться піктограма, яка ідентифікує міру як пов'язану з KPI. [16]

Для даного дослідження результатом KPI в розрізі відношення початкових фінансів до кінцевих результатів (активні та пасивні доходи) результат зображено на рис. 5.

	База	Норма	Ціль	Факт	Індекс KPI
Викладач	170	17000	85000	140460	834%
Хірург	1500	150000	750000	123290	82%
Помічник депутата	2500	250000	1250000	430750	173%

Рис.5. KPI

Досліджуючи отримані результати можна виділити, що переглядаючи стратегію для викладача ми бачимо, що показники у 8 разів перевищують заплановані цілі, хірург в свою чергу перевищив цілі всього у 0.8 разів, що відображає, що дії гравця з професією хірурга не до кінця коректні. Також досліджуючи третю стратегію з помічником депутата можемо побачити, що по розрахунку KPI маємо гарні показники, що в 1.7 разів перевищують заплановані.

# НУБІП України

## ВИСНОВКИ

В процесі виконання магістерського дослідження було досліджено предметну область з розглядом різномітних стратегій поведінки та вдосконалення їх до поданої тематики.

Оцінюючи актуальність роботи можемо визначити стан проблеми. На сьогодні немає систем які відобразять результати.

Очевидним результатом є спроектована база даних та розміщена у сховищі даних для аналізу та обробки захисту дипломних проектів та вивчення взаємозв'язків між учасниками навчального процесу в базі даних MS SQL Server. Була визначена структура сховища даних, створені вимірювання та розширений куб, куб заповнений даними за допомогою методів SSAS та SSIS при розгортанні в середовищі Visual Studio SSDT.

При розробці сховища даних була використана схема реляційної бази даних «КРИЖИЧКА», яка дозволяє мінімізувати надмірності даних і більш ефективно виконувати запити щодо структури вимірюваних величин.

Відповідно до завдання була визначена структура сховища даних і вимірювання, які потім відтворюються у багатовимірній таблиці та звітах, створених у Power BI, відповідають на всі запитання, визначені в завданні, і відображаються у звітах, діаграмах, гістограмах, моделі аналізу даних, які відображають якість відображення аналізованих даних. Нішове порівняння даних про стратегії та розрахованого результату KPI від показника максимальної очікуваної заробітної плати до показника середньої заробітної плати конкретного гравця і дозволяє розвивати аналітичну систему для інших стратегій.

# НУБІП України

# НУБІП України

## БІБЛІОГРАФІЧНИЙ ОПИС

1. 10 советов Роберта Кийосаки по созданию бизнеса university [Електронний ресурс]. – Режим доступу до ресурсу: <https://book24.ua/blog/10-sovetov-roberta-kiyosaki-po-sozdaniyu-biznesa/>

2. МОНИТОРИНГ ТА АНАЛІЗ УСПІШНОСТІ СТРАТЕГІЇ ПОВЕДІНКИ НА ПРИКЛАДІ ЕКОНОМІЧНОГО СИМУЛЯТОРА «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА» [Електронний ресурс]. – Режим доступу до ресурсу: <https://drive.google.com/file/d/1h9osGq77RwROGzfgpTwBYza12k6PmWgD/view?usp=sharing>

3. АНАЛІЗ УСПІШНОСТІ СТРАТЕГІЇ ПОВЕДІНКИ НА ПРИКЛАДІ ЕКОНОМІЧНОГО СИМУЛЯТОРА тези були представлені в рамках: IV Всеукраїнської науково-практичної інтернет конференції студентів аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2021», [Електронний ресурс]. – Режим доступу до ресурсу:

<https://drive.google.com/file/d/1wDc5EXRBBgR3slluQ3FCIbSTtY588gWb/view?usp=sharing>

4. CASHFLOW Classic [Електронний ресурс] режим доступу <https://www.richdad.com/products/cashflow-classic> – Назва з екрану.

5. CashGo [Електронний ресурс] режим доступу <https://cashgo.ru/levels/info> – Назва з екрану.

6. UML-моделирование [Електронний ресурс] режим доступу <https://fingers.by/ulcg/uml>

7. Діаграма прецедентів. Роль прецедентів при розробці ПС. Види прецедентів і відносини між ними. Правила опису [Електронний ресурс] режим доступу <http://um.co.ua/8/8-2/8-213194.html>

# НУБІП України

8. Що таке діаграма діяльності? - визначення з технології - Розвиток  
 [Електронний ресурс] режим доступу

<https://uk.theastrology.com/activity-diagram>

9. ЗАСТОСУВАННЯ UML (ЧАСТИНА 2). ДІАГРАМА

ПОСЛІДОВНОСТІ - SEQUENCE DIAGRAM [Електронний ресурс] режим

доступу [http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-](http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-)

[2-diagrama-poslidovnosti---sequence-diagram kafedra-kompyuternih-nauk-ta-](http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-)

[informaciynih-tehnologiy](http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy)

10. 12 правил Кодда [Електронний ресурс] режим доступу

<https://codelessons.com/tutorials/akademicheskii/uchim-subd/12-pravil-kodda>

11. СХОВИЩЕ ДАНИХ [Електронний ресурс] режим доступу

<https://biglib.info/5110-shovische-daniv.html>

12. Типи схем у моделюванні сховища даних - схема Star & Snowflake

[Електронний ресурс] режим доступу [https://uk.myservername.com/schema-](https://uk.myservername.com/schema-types-data-warehouse-modeling-star-snowflake-schema)

[types-data-warehouse-modeling-star-snowflake-schema](https://uk.myservername.com/schema-types-data-warehouse-modeling-star-snowflake-schema)

13. Нотатки про Business Intelligence - [Електронний ресурс] режим

доступу <http://dss-bi.com.ua/WP/tag/data-mining/>

14. Data Flow Task [Електронний ресурс] режим доступу

[https://docs.microsoft.com/en-us/sql/integration-services/control-flow/data-flow-](https://docs.microsoft.com/en-us/sql/integration-services/control-flow/data-flow-task?view=sql-server-ver15)

[task?view=sql-server-ver15](https://docs.microsoft.com/en-us/sql/integration-services/control-flow/data-flow-task?view=sql-server-ver15)

15. Как рассчитывается процентная ставка по кредиту? [Електронний

ресурс] режим доступу [https://mywallet.ua/blog/vse-o-kreditah/kak-](https://mywallet.ua/blog/vse-o-kreditah/kak-rasschityvaetsya-protstentnaya-stavka-po-kreditu/)

[rasschityvaetsya-protstentnaya-stavka-po-kreditu/](https://mywallet.ua/blog/vse-o-kreditah/kak-rasschityvaetsya-protstentnaya-stavka-po-kreditu/)

16. KPIs in tabular models [Електронний ресурс] режим доступу

<https://docs.microsoft.com/en-us/analysis-services/tabular-models/kpis-ssas->

[tabular?view=asallproducts-allversions](https://docs.microsoft.com/en-us/analysis-services/tabular-models/kpis-ssas-tabular?view=asallproducts-allversions)

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП ДОДАТКИ України

НУБІП України

НУБІП України

НУБІП України

НУБІП ДОДАТОК А України

НУБІП України

НУБІП України  
СТВОРЕННЯ БД І ТАБЛИЦЬ

НУБІП України

НУБІП України

НУБІП України

НУБІП КИЇВ 2021 України



INSERT INTO SessionLogFact VALUES (1, 5, 3, 170, 0, 170, 110, 60, 0, 0, 100, 0)  
INSERT INTO SessionLogFact VALUES (2, 1, 3, 170, 0, 170, 110, 60, 0, 0, 100, 0)  
INSERT INTO SessionLogFact VALUES (3, 1, 4, 170, 0, 170, 110, 60, 60, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (4, 1, 1, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (5, 1, 2, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (6, 1, 6, 0, 0, 0, 110, -110, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (7, 1, 7, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (8, 1, 6, 0, 0, 0, 110, -110, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (9, 1, 1, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (10, 1, 1, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (11, 1, 3, 170, 0, 170, 110, 60, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (12, 1, 6, 0, 0, 0, 110, -110, 0, 0, 160, 0)  
INSERT INTO SessionLogFact VALUES (13, 1, 5, 0, 0, 0, 150, -150, 1000, 0, 1160, 0)  
INSERT INTO SessionLogFact VALUES (14, 1, 2, 0, 0, 0, 150, -150, 0, 0, 1160, 0)  
INSERT INTO SessionLogFact VALUES (15, 1, 1, 0, 200, 200, 150, 50, 0, 500, 660, 0)  
INSERT INTO SessionLogFact VALUES (16, 1, 5, 0, 200, 200, 190, 10, 1000, 0, 1660, 0)  
INSERT INTO SessionLogFact VALUES (17, 1, 4, 0, 200, 200, 190, 10, 10, 0, 1670, 0)  
INSERT INTO SessionLogFact VALUES (18, 1, 1, 0, 200, 200, 190, 10, 0, 0, 1670, 0)  
INSERT INTO SessionLogFact VALUES (19, 1, 3, 0, 200, 200, 190, 10, 0, 0, 1670, 0)  
INSERT INTO SessionLogFact VALUES (20, 1, 5, 0, 200, 200, 230, -30, 1000, 0, 2670, 0)  
INSERT INTO SessionLogFact VALUES (21, 1, 4, 0, 200, 200, 230, -30, 0, 30, 2640, 0)  
INSERT INTO SessionLogFact VALUES (22, 1, 1, 0, 200, 200, 230, -30, 250, 0, 2890, 0)  
INSERT INTO SessionLogFact VALUES (23, 1, 5, 0, 200, 200, 270, -70, 1000, 0, 3890, 0)  
INSERT INTO SessionLogFact VALUES (24, 1, 7, 170, 200, 370, 270, 100, 0, 0, 3890, 0)  
INSERT INTO SessionLogFact VALUES (25, 1, 4, 170, 200, 370, 270, 100, 100, 0, 3990, 0)  
INSERT INTO SessionLogFact VALUES (26, 1, 4, 170, 200, 370, 270, 100, 100, 0, 4090, 0)  
INSERT INTO SessionLogFact VALUES (27, 1, 1, 170, 200, 370, 270, 100, 50, 0, 4140, 0)  
INSERT INTO SessionLogFact VALUES (28, 1, 3, 170, 200, 370, 270, 100, 0, 0, 4140, 0)  
INSERT INTO SessionLogFact VALUES (29, 1, 6, 0, 200, 200, 270, -70, 0, 0, 4140, 0)  
INSERT INTO SessionLogFact VALUES (30, 1, 4, 0, 200, 200, 270, -70, 0, 70, 4070, 0)  
INSERT INTO SessionLogFact VALUES (31, 1, 2, 0, 200, 200, 270, -70, 0, 0, 4070, 0)  
INSERT INTO SessionLogFact VALUES (32, 1, 4, 0, 200, 200, 270, -70, 0, 70, 4000, 0)  
INSERT INTO SessionLogFact VALUES (33, 1, 3, 0, 200, 200, 270, -70, 0, 0, 4000, 0)  
INSERT INTO SessionLogFact VALUES (34, 1, 7, 170, 200, 370, 270, 100, 0, 0, 4000, 0)  
INSERT INTO SessionLogFact VALUES (35, 1, 1, 170, 200, 370, 270, 100, 0, 3600, 400, 0)  
INSERT INTO SessionLogFact VALUES (36, 1, 5, 170, 200, 370, 310, 60, 1000, 0, 1400, 0)  
INSERT INTO SessionLogFact VALUES (37, 1, 3, 170, 200, 370, 310, 60, 0, 0, 1400, 0)  
INSERT INTO SessionLogFact VALUES (38, 1, 2, 170, 200, 370, 310, 60, 0, 0, 1400, 0)  
INSERT INTO SessionLogFact VALUES (39, 1, 6, 0, 200, 200, 310, -110, 0, 0, 1400, 0)  
INSERT INTO SessionLogFact VALUES (40, 1, 2, 0, 200, 200, 310, -110, 0, 500, 900, 0)  
INSERT INTO SessionLogFact VALUES (41, 1, 1, 0, 200, 200, 310, -110, 0, 0, 900, 0)

INSERT INTO SessionLogFact VALUES (42, 1, 1, 0, 300, 300, 310, -10, 0, 100, 800, 0)  
INSERT INTO SessionLogFact VALUES (43, 1, 3, 0, 300, 300, 310, -10, 0, 0, 800, 0)  
INSERT INTO SessionLogFact VALUES (44, 1, 3, 0, 300, 300, 310, -10, 0, 0, 800, 0)  
INSERT INTO SessionLogFact VALUES (45, 1, 4, 0, 300, 300, 310, -10, 0, 10, 790, 0)  
INSERT INTO SessionLogFact VALUES (46, 1, 1, 0, 500, 500, 310, 190, 0, 300, 490, 0)  
INSERT INTO SessionLogFact VALUES (47, 1, 7, 170, 500, 670, 310, 360, 0, 0, 490, 0)  
INSERT INTO SessionLogFact VALUES (48, 1, 6, 0, 500, 500, 310, 190, 0, 0, 490, 0)  
INSERT INTO SessionLogFact VALUES (49, 1, 7, 170, 500, 670, 310, 360, 0, 0, 490, 0)  
INSERT INTO SessionLogFact VALUES (50, 1, 6, 0, 500, 500, 310, 190, 0, 0, 490, 0)  
INSERT INTO SessionLogFact VALUES (51, 1, 1, 0, 500, 500, 310, 190, 0, 0, 490, 0)  
INSERT INTO SessionLogFact VALUES (52, 1, 1, 0, 600, 600, 310, 290, 0, 300, 190, 0)  
INSERT INTO SessionLogFact VALUES (53, 1, 3, 0, 600, 600, 310, 290, 0, 0, 190, 0)  
INSERT INTO SessionLogFact VALUES (54, 1, 7, 170, 600, 770, 310, 460, 0, 0, 190, 0)  
INSERT INTO SessionLogFact VALUES (55, 1, 5, 170, 600, 770, 350, 420, 1000, 0, 1190, 0)  
INSERT INTO SessionLogFact VALUES (56, 1, 2, 170, 600, 770, 350, 420, 0, 10, 1180, 0)  
INSERT INTO SessionLogFact VALUES (57, 1, 1, 170, 600, 770, 350, 420, 100, 0, 1280, 0)  
INSERT INTO SessionLogFact VALUES (58, 1, 5, 170, 600, 770, 390, 380, 1000, 0, 2280, 0)  
INSERT INTO SessionLogFact VALUES (59, 1, 4, 170, 600, 770, 390, 380, 380, 0, 2660, 0)  
INSERT INTO SessionLogFact VALUES (60, 1, 6, 0, 600, 600, 390, 210, 0, 0, 2660, 0)  
INSERT INTO SessionLogFact VALUES (61, 1, 7, 170, 600, 770, 390, 380, 0, 0, 2660, 0)  
INSERT INTO SessionLogFact VALUES (62, 1, 5, 170, 600, 770, 430, 340, 1000, 0, 3660, 0)  
INSERT INTO SessionLogFact VALUES (63, 1, 4, 170, 600, 770, 430, 340, 340, 0, 4000, 0)  
INSERT INTO SessionLogFact VALUES (64, 1, 1, 170, 1100, 1270, 430, 840, 0, 3000, 1000, 0)  
INSERT INTO SessionLogFact VALUES (65, 1, 5, 170, 1100, 1270, 470, 800, 1000, 0, 2000, 0)  
INSERT INTO SessionLogFact VALUES (66, 1, 3, 170, 1100, 1270, 470, 800, 0, 0, 2000, 0)  
INSERT INTO SessionLogFact VALUES (67, 1, 4, 170, 1100, 1270, 470, 800, 800, 0, 2800, 0)  
INSERT INTO SessionLogFact VALUES (68, 1, 4, 170, 1100, 1270, 470, 800, 800, 0, 3600, 0)  
INSERT INTO SessionLogFact VALUES (69, 1, 3, 170, 1100, 1270, 470, 800, 0, 0, 3600, 0)  
INSERT INTO SessionLogFact VALUES (70, 1, 1, 170, 1300, 1470, 470, 1000, 0, 800, 2800, 0)  
INSERT INTO SessionLogFact VALUES (71, 1, 6, 0, 1300, 1300, 470, 830, 0, 0, 2800, 0)  
INSERT INTO SessionLogFact VALUES (72, 1, 4, 0, 1300, 1300, 470, 830, 830, 0, 3630, 0)  
INSERT INTO SessionLogFact VALUES (73, 1, 2, 0, 1300, 1300, 470, 830, 0, 10, 3620, 0)  
INSERT INTO SessionLogFact VALUES (74, 1, 4, 0, 1300, 1300, 470, 830, 830, 0, 4450, 0)  
INSERT INTO SessionLogFact VALUES (75, 1, 3, 0, 1300, 1300, 470, 830, 0, 0, 4450, 0)  
INSERT INTO SessionLogFact VALUES (76, 1, 7, 170, 1300, 1470, 470, 1000, 0, 0, 4450, 0)  
INSERT INTO SessionLogFact VALUES (77, 1, 1, 170, 1300, 1470, 470, 1000, 0, 0, 4450, 0)  
INSERT INTO SessionLogFact VALUES (78, 1, 5, 170, 1300, 1470, 510, 960, 1000, 0, 5450, 0)  
INSERT INTO SessionLogFact VALUES (79, 1, 3, 170, 1300, 1470, 510, 960, 0, 0, 5450, 0)  
INSERT INTO SessionLogFact VALUES (80, 1, 2, 170, 1300, 1470, 510, 960, 0, 100, 5350, 0)  
INSERT INTO SessionLogFact VALUES (81, 1, 6, 0, 1300, 1300, 510, 790, 0, 0, 5350, 0)  
INSERT INTO SessionLogFact VALUES (82, 1, 2, 0, 1300, 1300, 510, 790, 0, 100, 5250, 0)

INSERT INTO SessionLogFact VALUES (83, 1, 1, 0, 1300, 1300, 510, 790, 0, 0, 5250, 0)  
INSERT INTO SessionLogFact VALUES (84, 1, 1, 0, 1400, 1400, 510, 890, 0, 100, 5150, 0)  
INSERT INTO SessionLogFact VALUES (85, 1, 3, 0, 1800, 1800, 510, 1290, 0, 0, 5150, 0)  
INSERT INTO SessionLogFact VALUES (86, 1, 7, 170, 1800, 1970, 510, 1460, 0, 0, 5150, 0)  
INSERT INTO SessionLogFact VALUES (87, 1, 1, 170, 2100, 2270, 510, 1760, 0, 1200, 3950, 0)  
INSERT INTO SessionLogFact VALUES (88, 1, 5, 170, 2100, 2270, 550, 1720, 1000, 0, 4950, 0)  
INSERT INTO SessionLogFact VALUES (89, 1, 3, 170, 2100, 2270, 550, 1720, 0, 0, 4950, 0)  
INSERT INTO SessionLogFact VALUES (90, 1, 2, 170, 2100, 2270, 550, 1720, 0, 1000, 3950, 0)  
INSERT INTO SessionLogFact VALUES (91, 1, 6, 0, 2100, 2100, 550, 1550, 0, 0, 3950, 0)  
INSERT INTO SessionLogFact VALUES (92, 1, 2, 0, 2100, 2100, 550, 1550, 0, 20, 3930, 0)  
INSERT INTO SessionLogFact VALUES (93, 1, 1, 0, 2100, 2100, 550, 1550, 300, 0, 4230, 0)  
INSERT INTO SessionLogFact VALUES (94, 1, 1, 0, 2100, 2100, 550, 1550, 0, 0, 4230, 0)  
INSERT INTO SessionLogFact VALUES (95, 1, 3, 0, 2700, 2700, 550, 2150, 0, 0, 4230, 0)  
INSERT INTO SessionLogFact VALUES (96, 1, 3, 0, 2700, 2700, 550, 2150, 0, 0, 4230, 0)  
INSERT INTO SessionLogFact VALUES (97, 1, 4, 0, 2700, 2700, 550, 2150, 2150, 0, 6380, 0)  
INSERT INTO SessionLogFact VALUES (98, 1, 1, 0, 2700, 2700, 550, 2150, 0, 600, 5780, 0)  
INSERT INTO SessionLogFact VALUES (99, 1, 7, 170, 2700, 2870, 550, 2320, 0, 0, 5780, 0)  
INSERT INTO SessionLogFact VALUES (100, 1, 6, 0, 2700, 2700, 550, 2150, 0, 0, 5780, 0)  
INSERT INTO SessionLogFact VALUES (101, 1, 7, 170, 2700, 2870, 550, 2320, 0, 0, 5780, 0)  
INSERT INTO SessionLogFact VALUES (102, 1, 6, 0, 2700, 2700, 550, 2150, 0, 0, 5780, 0)  
INSERT INTO SessionLogFact VALUES (103, 1, 1, 0, 2700, 2700, 550, 2150, 0, 1000, 4780, 0)  
INSERT INTO SessionLogFact VALUES (104, 1, 1, 0, 2800, 2800, 550, 2250, 0, 400, 4380, 0)  
INSERT INTO SessionLogFact VALUES (105, 1, 3, 0, 2800, 2800, 550, 2250, 0, 0, 4380, 0)  
INSERT INTO SessionLogFact VALUES (106, 1, 7, 170, 2800, 2970, 550, 2420, 0, 0, 4380, 0)  
INSERT INTO SessionLogFact VALUES (107, 1, 3, 170, 2800, 2970, 550, 2420, 0, 0, 4380, 0)  
INSERT INTO SessionLogFact VALUES (108, 1, 3, 170, 2800, 2970, 550, 2420, 0, 0, 4380, 0)  
INSERT INTO SessionLogFact VALUES (109, 1, 4, 170, 2800, 2970, 550, 2420, 2420, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (110, 1, 1, 170, 2800, 2970, 550, 2420, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (111, 1, 3, 170, 2800, 2970, 550, 2420, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (112, 1, 6, 0, 2800, 2800, 550, 2250, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (113, 1, 7, 170, 2800, 2970, 550, 2420, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (114, 1, 6, 0, 2800, 2800, 550, 2250, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (115, 1, 1, 0, 2800, 2800, 550, 2250, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (116, 1, 1, 0, 2800, 2800, 550, 2250, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (117, 1, 3, 0, 2800, 2800, 550, 2250, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (118, 1, 7, 170, 2800, 2970, 550, 2420, 0, 0, 6800, 0)  
INSERT INTO SessionLogFact VALUES (119, 1, 5, 170, 2800, 2970, 590, 2380, 1000, 0, 7800, 0)  
INSERT INTO SessionLogFact VALUES (120, 1, 2, 170, 2800, 2970, 590, 2380, 0, 4000, 3800, 0)  
INSERT INTO SessionLogFact VALUES (121, 1, 1, 170, 2800, 2970, 590, 2380, 0, 0, 3800, 0)  
INSERT INTO SessionLogFact VALUES (122, 1, 5, 170, 2800, 2970, 630, 2340, 1000, 0, 4800, 0)

INSERT INTO SessionLogFact VALUES (123, 1, 4, 170, 2800, 2970, 630, 2340, 2340, 0, 7140, 0)  
INSERT INTO SessionLogFact VALUES (124, 1, 6, 0, 2800, 2800, 630, 2170, 0, 0, 7140, 0)  
INSERT INTO SessionLogFact VALUES (125, 1, 7, 170, 2800, 2970, 630, 2340, 0, 0, 7140, 0)  
INSERT INTO SessionLogFact VALUES (126, 1, 5, 170, 2800, 2970, 670, 2300, 1000, 0, 8140, 0)  
INSERT INTO SessionLogFact VALUES (127, 1, 4, 170, 2800, 2970, 670, 2300, 2300, 0, 10440, 0)  
INSERT INTO SessionLogFact VALUES (128, 1, 1, 170, 3300, 3470, 670, 2800, 0, 3000, 7440, 0)  
INSERT INTO SessionLogFact VALUES (129, 1, 5, 170, 3300, 3470, 710, 2760, 1000, 0, 8440, 0)  
INSERT INTO SessionLogFact VALUES (130, 1, 3, 170, 3300, 3470, 710, 2760, 0, 0, 8440, 0)  
INSERT INTO SessionLogFact VALUES (131, 1, 4, 170, 3300, 3470, 710, 2760, 2760, 0, 11200, 0)  
INSERT INTO SessionLogFact VALUES (132, 1, 4, 170, 3300, 3470, 710, 2760, 2760, 0, 13960, 0)  
INSERT INTO SessionLogFact VALUES (133, 1, 1, 170, 3300, 3470, 710, 2760, 0, 900, 13060, 0)  
INSERT INTO SessionLogFact VALUES (134, 1, 3, 170, 3300, 3470, 710, 2760, 0, 0, 13060, 0)  
INSERT INTO SessionLogFact VALUES (135, 1, 6, 0, 3300, 3300, 710, 2590, 0, 0, 13060, 0)  
INSERT INTO SessionLogFact VALUES (136, 1, 4, 0, 3300, 3300, 710, 2590, 2590, 0, 15650, 0)  
INSERT INTO SessionLogFact VALUES (137, 1, 2, 0, 3300, 3300, 710, 2590, 0, 40, 15610, 0)  
INSERT INTO SessionLogFact VALUES (138, 1, 4, 0, 3300, 3300, 710, 2590, 2590, 0, 18200, 0)  
INSERT INTO SessionLogFact VALUES (139, 1, 3, 0, 3550, 3550, 710, 2840, 0, 0, 18200, 0)  
INSERT INTO SessionLogFact VALUES (140, 1, 3, 170, 3700, 3870, 710, 3160, 0, 0, 18200, 0)  
INSERT INTO SessionLogFact VALUES (141, 1, 1, 170, 3700, 3870, 710, 3160, 0, 6000, 12200, 0)  
INSERT INTO SessionLogFact VALUES (142, 1, 5, 170, 3700, 3870, 750, 3120, 1000, 0, 13200, 0)  
INSERT INTO SessionLogFact VALUES (143, 1, 3, 170, 3700, 3870, 750, 3120, 144000, 0, 157200, 0)  
INSERT INTO SessionLogFact VALUES (144, 1, 2, 170, 3700, 3870, 750, 3120, 0, 300, 156900, 0)  
INSERT INTO SessionLogFact VALUES (145, 1, 6, 0, 3700, 3700, 750, 2950, 0, 0, 156900, 0)  
INSERT INTO SessionLogFact VALUES (146, 1, 2, 0, 3700, 3700, 750, 2950, 0, 20, 156880, 0)  
INSERT INTO SessionLogFact VALUES (147, 1, 1, 0, 3700, 3700, 750, 2950, 50, 0, 156930, 0)  
INSERT INTO SessionLogFact VALUES (148, 1, 1, 0, 4300, 4300, 750, 3550, 0, 6000, 150930, 0)  
INSERT INTO SessionLogFact VALUES (149, 1, 3, 0, 4300, 4300, 750, 3550, 0, 0, 150930, 0)  
INSERT INTO SessionLogFact VALUES (150, 1, 3, 0, 4300, 4300, 750, 3550, 0, 0, 150930, 0)  
INSERT INTO SessionLogFact VALUES (151, 1, 4, 0, 4300, 4300, 750, 3550, 3550, 0, 154480, 0)  
INSERT INTO SessionLogFact VALUES (152, 1, 1, 0, 4300, 4300, 750, 3550, 0, 4000, 150480, 0)  
INSERT INTO SessionLogFact VALUES (153, 1, 7, 170, 4300, 4470, 750, 3720, 0, 0, 150480, 0)  
INSERT INTO SessionLogFact VALUES (154, 1, 6, 0, 4300, 4300, 750, 3550, 0, 0, 150480, 0)  
INSERT INTO SessionLogFact VALUES (155, 1, 7, 170, 4300, 4470, 750, 3720, 0, 0, 150480, 0)  
INSERT INTO SessionLogFact VALUES (156, 1, 6, 0, 4300, 4300, 750, 3550, 0, 0, 150480, 0)  
INSERT INTO SessionLogFact VALUES (157, 1, 1, 0, 4700, 4700, 750, 3950, 0, 1000, 149480, 0)  
INSERT INTO SessionLogFact VALUES (158, 1, 1, 0, 4700, 4700, 750, 3950, 300, 0, 149780, 0)  
INSERT INTO SessionLogFact VALUES (159, 1, 3, 0, 4700, 4700, 750, 3950, 0, 0, 149780, 0)  
INSERT INTO SessionLogFact VALUES (160, 1, 7, 170, 4700, 4870, 750, 4120, 0, 0, 149780, 0)  
INSERT INTO SessionLogFact VALUES (161, 1, 5, 170, 4700, 4870, 790, 4080, 1000, 0, 150780, 0)  
INSERT INTO SessionLogFact VALUES (162, 1, 2, 170, 4700, 4870, 790, 4080, 0, 50, 150730, 0)

INSERT INTO SessionLogFact VALUES (163, 1, 1, 170, 4700, 4870, 790, 4080, 0, 0, 150730, 0)  
INSERT INTO SessionLogFact VALUES (164, 1, 5, 170, 4700, 4870, 830, 4040, 1000, 0, 151730, 0)  
INSERT INTO SessionLogFact VALUES (165, 1, 4, 170, 4700, 4870, 830, 4040, 4040, 0, 155770, 0)  
INSERT INTO SessionLogFact VALUES (166, 1, 6, 0, 4700, 4700, 830, 3870, 0, 0, 155770, 0)  
INSERT INTO SessionLogFact VALUES (167, 1, 7, 170, 4700, 4870, 830, 4040, 0, 0, 155770, 0)  
INSERT INTO SessionLogFact VALUES (168, 1, 5, 170, 4700, 4870, 870, 4000, 1000, 0, 156770, 0)  
INSERT INTO SessionLogFact VALUES (169, 1, 4, 170, 4700, 4870, 870, 4000, 4000, 0, 160770, 0)  
INSERT INTO SessionLogFact VALUES (170, 1, 1, 170, 6700, 6870, 870, 6000, 0, 20000, 140770, 0)  
INSERT INTO SessionLogFact VALUES (171, 1, 5, 170, 6700, 6870, 910, 5960, 1000, 0, 141770, 0)  
INSERT INTO SessionLogFact VALUES (172, 1, 3, 170, 7000, 7170, 910, 6260, 0, 0, 141770, 0)  
INSERT INTO SessionLogFact VALUES (173, 1, 4, 170, 7000, 7170, 910, 6260, 6260, 0, 148030, 0)  
INSERT INTO SessionLogFact VALUES (174, 1, 4, 170, 7000, 7170, 910, 6260, 6260, 0, 154290, 0)  
INSERT INTO SessionLogFact VALUES (175, 1, 3, 170, 7000, 7170, 910, 6260, 0, 0, 154290, 0)  
INSERT INTO SessionLogFact VALUES (176, 1, 2, 170, 7000, 7170, 910, 6260, 0, 30, 154260, 0)  
INSERT INTO SessionLogFact VALUES (177, 1, 6, 0, 7000, 7000, 910, 6090, 0, 0, 154260, 0)  
INSERT INTO SessionLogFact VALUES (178, 1, 4, 0, 7000, 7000, 910, 6090, 6090, 0, 160350, 0)  
INSERT INTO SessionLogFact VALUES (179, 1, 2, 0, 7000, 7000, 910, 6090, 0, 20, 160330, 0)  
INSERT INTO SessionLogFact VALUES (180, 1, 4, 0, 7000, 7000, 910, 6090, 6090, 0, 166420, 0)  
INSERT INTO SessionLogFact VALUES (181, 1, 3, 0, 7000, 7000, 910, 6090, 0, 0, 166420, 0)  
INSERT INTO SessionLogFact VALUES (182, 1, 7, 170, 7000, 7170, 910, 6260, 0, 0, 166420, 0)  
INSERT INTO SessionLogFact VALUES (183, 1, 1, 170, 7000, 7170, 910, 6260, 0, 0, 166420, 0)  
INSERT INTO SessionLogFact VALUES (184, 1, 5, 170, 7000, 7170, 950, 6220, 1000, 0, 167420, 0)  
INSERT INTO SessionLogFact VALUES (185, 1, 3, 170, 7000, 7170, 950, 6220, 0, 0, 167420, 0)  
INSERT INTO SessionLogFact VALUES (186, 1, 2, 170, 7000, 7170, 950, 6220, 0, 300, 167120, 0)  
INSERT INTO SessionLogFact VALUES (187, 1, 6, 0, 7000, 7000, 950, 6050, 0, 0, 167120, 0)  
INSERT INTO SessionLogFact VALUES (188, 1, 2, 0, 7000, 7000, 950, 6050, 0, 20, 167100, 0)  
INSERT INTO SessionLogFact VALUES (189, 1, 1, 0, 7450, 7450, 950, 6500, 0, 90000, 77100, 0)  
INSERT INTO SessionLogFact VALUES (190, 1, 1, 0, 7450, 7450, 950, 6500, 0, 0, 77100, 0)  
INSERT INTO SessionLogFact VALUES (191, 1, 3, 0, 7450, 7450, 950, 6500, 0, 0, 77100, 0)  
INSERT INTO SessionLogFact VALUES (192, 1, 7, 170, 7450, 7620, 950, 6670, 0, 0, 77100, 0)  
INSERT INTO SessionLogFact VALUES (193, 1, 1, 170, 7950, 8120, 1500, 6620, 0, 25000, 52100, 55000)  
INSERT INTO SessionLogFact VALUES (194, 1, 5, 170, 7950, 8120, 1540, 6580, 0, 0, 52100, 55500)  
INSERT INTO SessionLogFact VALUES (195, 1, 3, 170, 7950, 8120, 990, 7130, 4000, 55500, 600, 0)  
INSERT INTO SessionLogFact VALUES (196, 1, 2, 170, 7950, 8120, 990, 7130, 0, 100, 500, 0)  
INSERT INTO SessionLogFact VALUES (197, 1, 6, 0, 7950, 7950, 990, 6960, 0, 0, 500, 0)  
INSERT INTO SessionLogFact VALUES (198, 1, 2, 0, 7950, 7950, 990, 6960, 0, 50, 450, 0)  
INSERT INTO SessionLogFact VALUES (199, 1, 1, 0, 7950, 7950, 990, 6960, 0, 0, 450, 0)  
INSERT INTO SessionLogFact VALUES (200, 1, 1, 0, 7950, 7950, 990, 6960, 0, 0, 450, 0)  
INSERT INTO SessionLogFact VALUES (201, 1, 3, 0, 7950, 7950, 990, 6960, 276000, 0, 276450, 0)  
INSERT INTO SessionLogFact VALUES (202, 1, 3, 0, 7950, 7950, 990, 6960, 0, 0, 276450, 0)

INSERT INTO SessionLogFact VALUES (203, 1, 4, 0, 7950, 7950, 990, 6960, 6960, 0, 283410, 0)  
INSERT INTO SessionLogFact VALUES (204, 1, 1, 0, 17950, 17950, 990, 16960, 0, 250000, 33410, 0)  
INSERT INTO SessionLogFact VALUES (205, 1, 7, 170, 17950, 18120, 990, 17130, 0, 0, 33410, 0)  
INSERT INTO SessionLogFact VALUES (206, 1, 6, 0, 17950, 17950, 990, 16960, 0, 0, 33410, 0)  
INSERT INTO SessionLogFact VALUES (207, 1, 7, 170, 17950, 18120, 990, 17130, 0, 0, 33410, 0)  
INSERT INTO SessionLogFact VALUES (208, 1, 6, 0, 17950, 17950, 990, 16960, 0, 0, 33410, 0)  
INSERT INTO SessionLogFact VALUES (209, 1, 1, 0, 18250, 18250, 990, 17260, 0, 1700, 31710, 0)  
INSERT INTO SessionLogFact VALUES (210, 1, 1, 0, 18250, 18250, 990, 17260, 0, 0, 31710, 0)  
INSERT INTO SessionLogFact VALUES (211, 1, 3, 0, 18250, 18250, 990, 17260, 0, 0, 31710, 0)  
INSERT INTO SessionLogFact VALUES (212, 1, 7, 170, 18250, 18420, 990, 17430, 0, 0, 31710, 0)  
INSERT INTO SessionLogFact VALUES (213, 1, 3, 170, 18950, 19120, 990, 18130, 0, 0, 31710, 0)  
INSERT INTO SessionLogFact VALUES (214, 1, 3, 170, 18950, 19120, 990, 18130, 0, 0, 31710, 0)  
INSERT INTO SessionLogFact VALUES (215, 1, 4, 170, 18950, 19120, 990, 18130, 18130, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (216, 1, 1, 170, 18950, 19120, 990, 18130, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (217, 1, 6, 0, 18950, 18950, 990, 17960, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (218, 1, 7, 170, 18950, 19120, 990, 18130, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (219, 1, 3, 170, 18950, 19120, 990, 18130, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (220, 1, 6, 0, 18950, 18950, 990, 17960, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (221, 1, 1, 0, 18950, 18950, 990, 17960, 0, 0, 49840, 0)  
INSERT INTO SessionLogFact VALUES (222, 1, 1, 0, 19100, 19100, 990, 18110, 0, 15000, 34840, 70000)  
INSERT INTO SessionLogFact VALUES (223, 1, 3, 0, 19100, 19100, 990, 18110, 0, 0, 34840, 70700)  
INSERT INTO SessionLogFact VALUES (224, 1, 7, 170, 19100, 19270, 990, 18280, 0, 0, 34840, 71400)  
INSERT INTO SessionLogFact VALUES (225, 1, 5, 170, 19100, 19270, 1130, 18140, 1000, 0, 35840, 72100)  
INSERT INTO SessionLogFact VALUES (226, 1, 2, 170, 19100, 19270, 1130, 18140, 0, 0, 35840, 72800)  
INSERT INTO SessionLogFact VALUES (227, 1, 1, 170, 19100, 19270, 1130, 18140, 0, 0, 35840, 73500)  
INSERT INTO SessionLogFact VALUES (228, 1, 5, 170, 19100, 19270, 1170, 18100, 1000, 0, 36840, 74200)  
INSERT INTO SessionLogFact VALUES (229, 1, 4, 170, 19100, 19270, 1170, 18100, 18100, 50000, 4940, 24900)  
INSERT INTO SessionLogFact VALUES (230, 1, 6, 0, 19100, 19100, 1170, 17930, 0, 0, 4940, 25600)  
INSERT INTO SessionLogFact VALUES (231, 1, 7, 170, 19100, 19270, 1170, 18100, 0, 0, 4940, 26300)  
INSERT INTO SessionLogFact VALUES (232, 1, 5, 170, 19100, 19270, 1210, 18060, 1000, 0, 5940, 27000)  
INSERT INTO SessionLogFact VALUES (233, 1, 4, 170, 19100, 19270, 1210, 18060, 18060, 0, 24000, 27700)  
INSERT INTO SessionLogFact VALUES (234, 1, 1, 170, 19100, 19270, 1210, 18060, 0, 10, 23990, 28400)  
INSERT INTO SessionLogFact VALUES (235, 1, 5, 170, 19100, 19270, 1250, 18020, 1000, 0, 24990, 29100)  
INSERT INTO SessionLogFact VALUES (236, 1, 6, 0, 19100, 19100, 1250, 17850, 0, 0, 24990, 29800)  
INSERT INTO SessionLogFact VALUES (237, 1, 4, 0, 19100, 19100, 1250, 17850, 17850, 0, 42840, 30500)  
INSERT INTO SessionLogFact VALUES (238, 1, 4, 0, 19100, 19100, 1250, 17850, 17850, 0, 60690, 31200)  
INSERT INTO SessionLogFact VALUES (239, 1, 7, 170, 19100, 19270, 1250, 18020, 0, 31200, 29490, 0)  
INSERT INTO SessionLogFact VALUES (240, 1, 6, 0, 19100, 19100, 1250, 17850, 0, 0, 29490, 0)  
INSERT INTO SessionLogFact VALUES (241, 1, 7, 170, 19100, 19270, 1250, 18020, 0, 0, 29490, 0)  
INSERT INTO SessionLogFact VALUES (242, 1, 4, 170, 19100, 19270, 1250, 18020, 18020, 0, 47510, 0)

INSERT INTO SessionLogFact VALUES (243, 1, 2, 170, 19100, 19270, 1250, 18020, 0, 200, 47310, 0)  
INSERT INTO SessionLogFact VALUES (244, 1, 4, 170, 19100, 19270, 1250, 18020, 18020, 0, 65330, 0)  
INSERT INTO SessionLogFact VALUES (245, 1, 3, 170, 19100, 19270, 1250, 18020, 0, 0, 65330, 0)  
INSERT INTO SessionLogFact VALUES (246, 1, 6, 0, 19100, 19100, 1250, 17850, 0, 0, 65330, 0)  
INSERT INTO SessionLogFact VALUES (247, 1, 1, 0, 19450, 19450, 1250, 18200, 0, 15000, 50330, 60000)  
INSERT INTO SessionLogFact VALUES (248, 1, 5, 0, 19450, 19450, 1290, 18160, 1000, 0, 51330, 60600)  
INSERT INTO SessionLogFact VALUES (249, 1, 3, 170, 19450, 19620, 1290, 18330, 99000, 60600, 89730, 0)  
INSERT INTO SessionLogFact VALUES (250, 1, 2, 170, 19450, 19620, 1290, 18330, 0, 0, 89730, 0)  
INSERT INTO SessionLogFact VALUES (251, 1, 6, 0, 19450, 19450, 1290, 18160, 0, 0, 89730, 0)  
INSERT INTO SessionLogFact VALUES (252, 1, 2, 0, 19450, 19450, 1290, 18160, 0, 500, 89230, 0)  
INSERT INTO SessionLogFact VALUES (253, 1, 1, 0, 19450, 19450, 1290, 18160, 0, 0, 89230, 0)  
INSERT INTO SessionLogFact VALUES (254, 1, 1, 0, 19550, 19550, 1290, 18260, 0, 100, 89130, 0)  
INSERT INTO SessionLogFact VALUES (255, 1, 3, 0, 19550, 19550, 1290, 18260, 0, 0, 89130, 0)  
INSERT INTO SessionLogFact VALUES (256, 1, 3, 0, 19550, 19550, 1290, 18260, 0, 0, 89130, 0)  
INSERT INTO SessionLogFact VALUES (257, 1, 4, 0, 19550, 19550, 1290, 18260, 18260, 0, 107390, 0)  
INSERT INTO SessionLogFact VALUES (258, 1, 1, 0, 25550, 25550, 1290, 24260, 0, 100000, 7390, 0)  
INSERT INTO SessionLogFact VALUES (259, 1, 7, 170, 25550, 25720, 1290, 24430, 0, 0, 7390, 0)  
INSERT INTO SessionLogFact VALUES (260, 1, 6, 0, 25550, 25550, 1290, 24260, 0, 0, 7390, 0)  
INSERT INTO SessionLogFact VALUES (261, 1, 7, 170, 25550, 25720, 1290, 24430, 0, 0, 7390, 0)  
INSERT INTO SessionLogFact VALUES (262, 1, 6, 0, 25550, 25550, 1290, 24260, 0, 0, 7390, 0)  
INSERT INTO SessionLogFact VALUES (263, 1, 1, 0, 25550, 25550, 1290, 24260, 300, 0, 7690, 0)  
INSERT INTO SessionLogFact VALUES (264, 1, 1, 0, 25550, 25550, 1290, 24260, 0, 3000, 4690, 0)  
INSERT INTO SessionLogFact VALUES (265, 1, 3, 0, 25550, 25550, 1290, 24260, 0, 0, 4690, 0)  
INSERT INTO SessionLogFact VALUES (266, 1, 7, 170, 25550, 25720, 1290, 24430, 0, 0, 4690, 0)  
INSERT INTO SessionLogFact VALUES (267, 1, 5, 170, 25550, 25720, 1330, 24390, 1000, 0, 5690, 0)  
INSERT INTO SessionLogFact VALUES (268, 1, 2, 170, 25550, 25720, 1330, 24390, 0, 150, 5540, 0)  
INSERT INTO SessionLogFact VALUES (269, 1, 1, 170, 25550, 25720, 1330, 24390, 0, 10, 5530, 0)  
INSERT INTO SessionLogFact VALUES (270, 1, 5, 170, 25550, 25720, 1370, 24350, 1000, 0, 6530, 0)  
INSERT INTO SessionLogFact VALUES (271, 1, 4, 170, 25550, 25720, 1370, 24350, 24350, 0, 30880, 0)  
INSERT INTO SessionLogFact VALUES (272, 1, 6, 0, 25550, 25550, 1370, 24180, 0, 0, 30880, 0)  
INSERT INTO SessionLogFact VALUES (273, 1, 7, 170, 25550, 25720, 1370, 24350, 0, 0, 30880, 0)  
INSERT INTO SessionLogFact VALUES (274, 1, 5, 170, 25550, 25720, 1410, 24310, 1000, 0, 31880, 0)  
INSERT INTO SessionLogFact VALUES (275, 1, 4, 170, 25550, 25720, 1410, 24310, 24310, 0, 56190, 0)  
INSERT INTO SessionLogFact VALUES (276, 1, 1, 170, 25550, 25720, 1410, 24310, 0, 2500, 53690, 0)  
INSERT INTO SessionLogFact VALUES (277, 1, 5, 170, 25550, 25720, 1450, 24270, 1000, 0, 54690, 0)  
INSERT INTO SessionLogFact VALUES (278, 1, 6, 0, 25550, 25550, 1450, 24100, 0, 0, 54690, 0)  
INSERT INTO SessionLogFact VALUES (279, 1, 4, 0, 25550, 25550, 1450, 24100, 24100, 0, 78790, 0)  
INSERT INTO SessionLogFact VALUES (280, 1, 4, 0, 25550, 25550, 1450, 24100, 24100, 0, 102890, 0)  
INSERT INTO SessionLogFact VALUES (281, 1, 7, 170, 25550, 25720, 1450, 24270, 0, 0, 102890, 0)  
INSERT INTO SessionLogFact VALUES (282, 1, 3, 170, 25550, 25720, 1450, 24270, 0, 0, 102890, 0)

INSERT INTO SessionLogFact VALUES (283, 1, 6, 0, 25550, 25550, 1450, 24100, 0, 0, 102890, 0)  
INSERT INTO SessionLogFact VALUES (284, 1, 4, 0, 25550, 25550, 1450, 24100, 24100, 0, 126990, 0)  
INSERT INTO SessionLogFact VALUES (285, 1, 2, 0, 25550, 25550, 1450, 24100, 0, 80, 126910, 0)  
INSERT INTO SessionLogFact VALUES (286, 1, 4, 0, 25550, 25550, 1450, 24100, 24100, 0, 151010, 0)  
INSERT INTO SessionLogFact VALUES (287, 1, 3, 0, 25550, 25550, 1450, 24100, 0, 0, 151010, 0)  
INSERT INTO SessionLogFact VALUES (288, 1, 7, 170, 25550, 25720, 1450, 24270, 0, 0, 151010, 0)  
INSERT INTO SessionLogFact VALUES (289, 1, 1, 170, 26150, 26320, 1450, 24870, 0, 55000, 96010, 100000)  
INSERT INTO SessionLogFact VALUES (290, 1, 5, 170, 26150, 26320, 1490, 24830, 1000, 0, 97010, 101000)  
INSERT INTO SessionLogFact VALUES (291, 1, 3, 170, 26150, 26320, 1490, 24830, 0, 0, 97010, 102000)  
INSERT INTO SessionLogFact VALUES (292, 1, 2, 170, 26150, 26320, 1490, 24830, 0, 40, 96970, 103000)  
INSERT INTO SessionLogFact VALUES (293, 1, 6, 0, 26150, 26150, 1490, 24660, 0, 0, 96970, 104000)  
INSERT INTO SessionLogFact VALUES (294, 1, 2, 0, 26150, 26150, 1490, 24660, 0, 50, 96920, 105000)  
INSERT INTO SessionLogFact VALUES (295, 1, 1, 0, 26150, 26150, 1490, 24660, 0, 0, 96920, 106000)  
INSERT INTO SessionLogFact VALUES (296, 1, 1, 0, 31150, 31150, 1490, 29660, 0, 70000, 26920, 107000)  
INSERT INTO SessionLogFact VALUES (297, 1, 3, 0, 31150, 31150, 1490, 29660, 15000, 0, 41920, 108000)  
INSERT INTO SessionLogFact VALUES (298, 1, 6, 0, 31150, 31150, 1490, 29660, 0, 0, 41920, 109000)  
INSERT INTO SessionLogFact VALUES (299, 1, 1, 0, 31150, 31150, 1490, 29660, 0, 0, 41920, 110000)  
INSERT INTO SessionLogFact VALUES (300, 1, 5, 0, 31150, 31150, 1530, 29620, 1000, 0, 42920, 111000)  
INSERT INTO SessionLogFact VALUES (301, 1, 7, 170, 31150, 31320, 1530, 29790, 0, 0, 42920, 112000)  
INSERT INTO SessionLogFact VALUES (302, 1, 2, 170, 31150, 31320, 1530, 29790, 0, 300, 42620, 113000)  
INSERT INTO SessionLogFact VALUES (303, 1, 6, 0, 31150, 31150, 1530, 29620, 0, 0, 42620, 114000)  
INSERT INTO SessionLogFact VALUES (304, 1, 2, 0, 31150, 31150, 1530, 29620, 0, 100, 42520, 115000)  
INSERT INTO SessionLogFact VALUES (305, 1, 1, 0, 31150, 31150, 1530, 29620, 0, 0, 42520, 116000)  
INSERT INTO SessionLogFact VALUES (306, 1, 1, 0, 31150, 31150, 1530, 29620, 0, 0, 42520, 117000)  
INSERT INTO SessionLogFact VALUES (307, 1, 3, 0, 31150, 31150, 1530, 29620, 0, 0, 42520, 118000)  
INSERT INTO SessionLogFact VALUES (308, 1, 3, 0, 31150, 31150, 1530, 29620, 0, 0, 42520, 119000)  
INSERT INTO SessionLogFact VALUES (309, 1, 4, 0, 31150, 31150, 1530, 29620, 29620, 0, 72140, 120000)  
INSERT INTO SessionLogFact VALUES (310, 1, 1, 0, 31750, 31750, 1530, 30220, 0, 4000, 68140, 121000)  
INSERT INTO SessionLogFact VALUES (311, 1, 7, 170, 31750, 31920, 1530, 30390, 0, 0, 68140, 122000)  
INSERT INTO SessionLogFact VALUES (312, 1, 6, 0, 31750, 31750, 1530, 30220, 0, 0, 68140, 123000)  
INSERT INTO SessionLogFact VALUES (313, 1, 7, 170, 31750, 31920, 1530, 30390, 0, 0, 68140, 124000)  
INSERT INTO SessionLogFact VALUES (314, 1, 6, 0, 31750, 31750, 1530, 30220, 0, 0, 68140, 125000)  
INSERT INTO SessionLogFact VALUES (315, 1, 1, 0, 31750, 31750, 1530, 30220, 0, 0, 68140, 126000)  
INSERT INTO SessionLogFact VALUES (316, 1, 1, 0, 31750, 31750, 1530, 30220, 0, 2400, 65740, 127000)  
INSERT INTO SessionLogFact VALUES (317, 1, 3, 0, 31750, 31750, 1530, 30220, 0, 0, 65740, 128000)  
INSERT INTO SessionLogFact VALUES (318, 1, 7, 170, 31750, 31920, 1530, 30390, 0, 0, 65740, 129000)  
INSERT INTO SessionLogFact VALUES (319, 1, 3, 170, 31850, 32020, 1530, 30490, 0, 0, 65740, 130000)  
INSERT INTO SessionLogFact VALUES (320, 1, 3, 170, 31850, 32020, 1530, 30490, 100, 0, 65840, 131000)  
INSERT INTO SessionLogFact VALUES (321, 1, 4, 170, 31850, 32020, 1530, 30490, 30490, 0, 96330, 132000)  
INSERT INTO SessionLogFact VALUES (322, 1, 1, 170, 32150, 32320, 1530, 30790, 0, 900, 95430, 133000)



INSERT INTO SessionLogFact VALUES (323, 1, 6, 0, 31250, 31250, 1530, 29720, 0, 0, 95430, 134000)  
INSERT INTO SessionLogFact VALUES (324, 1, 7, 170, 31250, 31420, 1530, 29890, 0, 0, 95430, 135000)  
INSERT INTO SessionLogFact VALUES (325, 1, 3, 170, 31250, 31420, 1530, 29890, 0, 0, 95430, 136000)  
INSERT INTO SessionLogFact VALUES (326, 1, 6, 0, 31250, 31250, 1530, 29720, 0, 0, 95430, 137000)  
INSERT INTO SessionLogFact VALUES (327, 1, 1, 0, 31550, 31550, 1530, 30020, 0, 800, 94630, 138000)  
INSERT INTO SessionLogFact VALUES (328, 1, 1, 0, 31550, 31550, 1530, 30020, 0, 0, 94630, 139000)  
INSERT INTO SessionLogFact VALUES (329, 1, 3, 0, 31550, 31550, 1530, 30020, 0, 0, 94630, 140000)  
INSERT INTO SessionLogFact VALUES (330, 1, 7, 170, 31550, 31720, 1530, 30190, 0, 0, 94630, 141000)  
INSERT INTO SessionLogFact VALUES (331, 1, 5, 170, 31550, 31720, 1570, 30150, 1000, 0, 95630, 142000)  
INSERT INTO SessionLogFact VALUES (332, 1, 2, 170, 31550, 31720, 1570, 30150, 0, 200, 95430, 143000)  
INSERT INTO SessionLogFact VALUES (333, 1, 1, 170, 31550, 31720, 1570, 30150, 0, 6000, 89430, 144000)  
INSERT INTO SessionLogFact VALUES (334, 1, 5, 170, 31550, 31720, 1610, 30110, 1000, 0, 90430, 145000)  
INSERT INTO SessionLogFact VALUES (335, 1, 4, 170, 31550, 31720, 1610, 30110, 30110, 0, 120540, 146000)  
INSERT INTO SessionLogFact VALUES (336, 1, 4, 170, 31550, 31720, 1610, 30110, 30110, 146000, 4650, 0)  
INSERT INTO SessionLogFact VALUES (337, 1, 6, 0, 31550, 31550, 1610, 29940, 0, 0, 4650, 0)  
INSERT INTO SessionLogFact VALUES (338, 1, 5, 0, 31550, 31550, 1650, 29900, 1000, 0, 5650, 0)  
INSERT INTO SessionLogFact VALUES (339, 1, 4, 0, 31550, 31550, 1650, 29900, 29900, 0, 35550, 0)  
INSERT INTO SessionLogFact VALUES (340, 1, 1, 0, 31750, 31750, 1650, 30100, 0, 800, 34750, 0)  
INSERT INTO SessionLogFact VALUES (341, 1, 5, 0, 31750, 31750, 1690, 30060, 1000, 0, 35750, 0)  
INSERT INTO SessionLogFact VALUES (342, 1, 7, 170, 31750, 31920, 1690, 30230, 0, 0, 35750, 0)  
INSERT INTO SessionLogFact VALUES (343, 1, 4, 170, 31750, 31920, 1690, 30230, 30230, 0, 65980, 0)  
INSERT INTO SessionLogFact VALUES (344, 1, 4, 170, 31750, 31920, 1690, 30230, 30230, 0, 96210, 0)  
INSERT INTO SessionLogFact VALUES (345, 1, 6, 0, 31750, 31750, 1690, 30060, 0, 0, 96210, 0)  
INSERT INTO SessionLogFact VALUES (346, 1, 7, 170, 31750, 31920, 1690, 30230, 0, 0, 96210, 0)  
INSERT INTO SessionLogFact VALUES (347, 1, 6, 0, 31750, 31750, 1690, 30060, 0, 0, 96210, 0)  
INSERT INTO SessionLogFact VALUES (348, 1, 4, 0, 31750, 31750, 1690, 30060, 30060, 0, 126270, 0)  
INSERT INTO SessionLogFact VALUES (349, 1, 2, 0, 31750, 31750, 1690, 30060, 0, 700, 125570, 0)  
INSERT INTO SessionLogFact VALUES (350, 1, 4, 0, 31750, 31750, 1690, 30060, 30060, 0, 155630, 0)  
INSERT INTO SessionLogFact VALUES (351, 1, 3, 0, 31750, 31750, 1690, 30060, 0, 0, 155630, 0)  
INSERT INTO SessionLogFact VALUES (352, 1, 7, 170, 31750, 31920, 1690, 30230, 0, 0, 155630, 0)  
INSERT INTO SessionLogFact VALUES (353, 1, 1, 170, 32400, 32570, 1690, 30880, 0, 40000, 115630, 120000)  
INSERT INTO SessionLogFact VALUES (354, 1, 5, 170, 32400, 32570, 1730, 30840, 1000, 0, 116630, 121200)  
INSERT INTO SessionLogFact VALUES (355, 1, 6, 0, 32400, 32400, 1730, 30670, 0, 0, 116630, 122400)  
INSERT INTO SessionLogFact VALUES (356, 1, 2, 0, 32400, 32400, 1730, 30670, 0, 40, 116590, 123600)  
INSERT INTO SessionLogFact VALUES (357, 1, 7, 170, 32400, 32570, 1730, 30840, 0, 0, 116590, 124800)  
INSERT INTO SessionLogFact VALUES (358, 1, 2, 170, 32400, 32570, 1730, 30840, 0, 500, 116090, 126000)  
INSERT INTO SessionLogFact VALUES (359, 1, 1, 170, 35900, 36070, 1730, 34340, 0, 50000, 66090, 127200)  
INSERT INTO SessionLogFact VALUES (360, 1, 1, 170, 35900, 36070, 1730, 34340, 0, 30000, 36090, 128400)  
INSERT INTO SessionLogFact VALUES (361, 1, 3, 170, 35900, 36070, 1730, 34340, 0, 0, 36090, 129600)  
INSERT INTO SessionLogFact VALUES (362, 1, 3, 170, 35900, 36070, 1730, 34340, 42000, 0, 78090, 130800)

INSERT INTO SessionLogFact VALUES (363, 1, 4, 170, 35900, 36070, 1730, 34340, 34340, 0, 112430, 132000)  
INSERT INTO SessionLogFact VALUES (364, 1, 1, 170, 35900, 36070, 1730, 34340, 0, 0, 112430, 133200)  
INSERT INTO SessionLogFact VALUES (365, 1, 6, 0, 35900, 35900, 1730, 34170, 0, 0, 112430, 134400)  
INSERT INTO SessionLogFact VALUES (366, 1, 7, 170, 35900, 36070, 1730, 34340, 0, 0, 112430, 135600)  
INSERT INTO SessionLogFact VALUES (367, 1, 3, 170, 35900, 36070, 1730, 34340, 0, 0, 112430, 136800)  
INSERT INTO SessionLogFact VALUES (368, 1, 6, 0, 35900, 35900, 1730, 34170, 0, 0, 112430, 138000)  
INSERT INTO SessionLogFact VALUES (369, 1, 1, 0, 37400, 37400, 1730, 35670, 0, 20000, 92430, 139200)  
INSERT INTO SessionLogFact VALUES (370, 1, 1, 0, 37400, 37400, 1730, 35670, 0, 0, 92430, 140400)  
INSERT INTO SessionLogFact VALUES (371, 1, 3, 0, 37400, 37400, 1730, 35670, 0, 0, 92430, 141600)  
INSERT INTO SessionLogFact VALUES (372, 1, 7, 170, 37400, 37570, 1730, 35840, 0, 0, 92430, 142800)  
INSERT INTO SessionLogFact VALUES (373, 1, 5, 170, 37400, 37570, 1770, 35800, 1000, 0, 93430, 144000)  
INSERT INTO SessionLogFact VALUES (374, 1, 2, 170, 37400, 37570, 1770, 35800, 0, 50, 93380, 145200)  
INSERT INTO SessionLogFact VALUES (375, 1, 1, 170, 37400, 37570, 1770, 35800, 0, 0, 93380, 146400)  
INSERT INTO SessionLogFact VALUES (376, 1, 5, 170, 37400, 37570, 1810, 35760, 1000, 0, 94380, 147600)  
INSERT INTO SessionLogFact VALUES (377, 1, 4, 170, 37400, 37570, 1810, 35760, 35760, 0, 130140, 148800)  
INSERT INTO SessionLogFact VALUES (378, 1, 6, 0, 37400, 37400, 1810, 35590, 0, 0, 130140, 150000)  
INSERT INTO SessionLogFact VALUES (379, 1, 7, 170, 37400, 37570, 1810, 35760, 0, 0, 130140, 151200)  
INSERT INTO SessionLogFact VALUES (380, 1, 5, 170, 37400, 37570, 1850, 35720, 1000, 0, 131140, 152400)  
INSERT INTO SessionLogFact VALUES (381, 1, 4, 170, 37400, 37570, 1850, 35720, 35720, 0, 166860, 153600)  
INSERT INTO SessionLogFact VALUES (382, 1, 1, 170, 37400, 37570, 1850, 35720, 0, 25000, 141860, 154800)  
INSERT INTO SessionLogFact VALUES (383, 1, 5, 170, 37400, 37570, 1890, 35680, 1000, 0, 142860, 156000)  
INSERT INTO SessionLogFact VALUES (384, 1, 6, 0, 37400, 37400, 1890, 35510, 0, 0, 142860, 157200)  
INSERT INTO SessionLogFact VALUES (385, 1, 4, 0, 37400, 37400, 1890, 35510, 35510, 157200, 21170, 0)  
INSERT INTO SessionLogFact VALUES (386, 1, 4, 0, 37400, 37400, 1890, 35510, 35510, 0, 56680, 0)  
INSERT INTO SessionLogFact VALUES (387, 1, 7, 170, 37400, 37570, 1890, 35680, 0, 0, 56680, 0)  
INSERT INTO SessionLogFact VALUES (388, 1, 3, 170, 37400, 37570, 1890, 35680, 0, 0, 56680, 0)  
INSERT INTO SessionLogFact VALUES (389, 1, 6, 0, 37400, 37400, 1890, 35510, 0, 0, 56680, 0)  
INSERT INTO SessionLogFact VALUES (390, 1, 4, 0, 37400, 37400, 1890, 35510, 35510, 0, 92190, 0)  
INSERT INTO SessionLogFact VALUES (391, 1, 2, 0, 37400, 37400, 1890, 35510, 0, 150, 92040, 0)  
INSERT INTO SessionLogFact VALUES (392, 1, 4, 0, 37400, 37400, 1890, 35510, 35510, 0, 127550, 0)  
INSERT INTO SessionLogFact VALUES (393, 1, 3, 0, 37050, 37050, 1890, 35160, 130000, 0, 257550, 0)  
INSERT INTO SessionLogFact VALUES (394, 1, 7, 170, 37050, 37220, 1890, 35330, 0, 0, 257550, 0)  
INSERT INTO SessionLogFact VALUES (395, 1, 1, 170, 37500, 37670, 1890, 35780, 0, 80000, 177550, 0)  
INSERT INTO SessionLogFact VALUES (396, 1, 5, 170, 37500, 37670, 1930, 35740, 1000, 0, 178550, 0)  
INSERT INTO SessionLogFact VALUES (397, 1, 6, 0, 37500, 37500, 1930, 35570, 0, 0, 178550, 0)  
INSERT INTO SessionLogFact VALUES (398, 1, 2, 0, 37500, 37500, 1930, 35570, 0, 100, 178450, 0)  
INSERT INTO SessionLogFact VALUES (399, 1, 3, 0, 37500, 37500, 1930, 35570, 0, 0, 178450, 0)  
INSERT INTO SessionLogFact VALUES (400, 1, 2, 0, 37500, 37500, 1930, 35570, 0, 20, 178430, 0)  
INSERT INTO SessionLogFact VALUES (401, 1, 1, 0, 37500, 37500, 1930, 35570, 0, 0, 178430, 0)  
INSERT INTO SessionLogFact VALUES (402, 1, 1, 0, 37500, 37500, 1930, 35570, 0, 2500, 175930, 0)

INSERT INTO SessionLogFact VALUES (403, 1, 3, 0, 37500, 37500, 1930, 35570, 718750, 0, 894680, 0)  
INSERT INTO SessionLogFact VALUES (404, 1, 7, 170, 37500, 37670, 1930, 35740, 0, 0, 894680, 0)  
INSERT INTO SessionLogFact VALUES (405, 1, 1, 170, 37500, 37670, 1930, 35740, 0, 0, 894680, 0)  
INSERT INTO SessionLogFact VALUES (406, 1, 5, 170, 37500, 37670, 1970, 35700, 1000, 0, 895680, 0)  
INSERT INTO SessionLogFact VALUES (407, 1, 6, 0, 37500, 37500, 1970, 35530, 0, 0, 895680, 0)  
INSERT INTO SessionLogFact VALUES (408, 1, 2, 0, 37500, 37500, 1970, 35530, 0, 100, 895580, 0)  
INSERT INTO SessionLogFact VALUES (409, 1, 3, 0, 37500, 37500, 1970, 35530, 0, 0, 895580, 0)  
INSERT INTO SessionLogFact VALUES (410, 1, 2, 0, 37500, 37500, 1970, 35530, 0, 30, 895550, 0)  
INSERT INTO SessionLogFact VALUES (411, 1, 1, 0, 37800, 37800, 1970, 35830, 0, 1800, 893750, 0)  
INSERT INTO SessionLogFact VALUES (412, 1, 1, 0, 37800, 37800, 1970, 35830, 0, 0, 893750, 0)  
INSERT INTO SessionLogFact VALUES (413, 1, 3, 0, 37800, 37800, 1970, 35830, 0, 0, 893750, 0)  
INSERT INTO SessionLogFact VALUES (414, 1, 3, 0, 37800, 37800, 1970, 35830, 250000, 0, 1143750, 0)  
INSERT INTO SessionLogFact VALUES (415, 1, 4, 0, 37800, 37800, 1970, 35830, 35830, 0, 1179580, 0)  
INSERT INTO SessionLogFact VALUES (416, 1, 1, 0, 37800, 37800, 1970, 35830, 400, 0, 1179980, 0)  
INSERT INTO SessionLogFact VALUES (417, 1, 7, 170, 37800, 37970, 1970, 36000, 0, 0, 1179980, 0)  
INSERT INTO SessionLogFact VALUES (418, 1, 6, 0, 37800, 37800, 1970, 35830, 0, 0, 1179980, 0)  
INSERT INTO SessionLogFact VALUES (419, 1, 7, 170, 37800, 37970, 1970, 36000, 0, 0, 1179980, 0)  
INSERT INTO SessionLogFact VALUES (420, 1, 6, 0, 37800, 37800, 1970, 35830, 0, 0, 1179980, 0)  
INSERT INTO SessionLogFact VALUES (421, 1, 1, 0, 49800, 49800, 1970, 47830, 0, 300000, 879980, 0)  
INSERT INTO SessionLogFact VALUES (422, 1, 1, 0, 50200, 50200, 1970, 48230, 0, 85000, 794980, 0)  
INSERT INTO SessionLogFact VALUES (423, 1, 3, 0, 50200, 50200, 1970, 48230, 0, 0, 794980, 0)  
INSERT INTO SessionLogFact VALUES (424, 1, 3, 0, 50200, 50200, 1970, 48230, 0, 0, 794980, 0)  
INSERT INTO SessionLogFact VALUES (425, 1, 3, 0, 50200, 50200, 1970, 48230, 0, 0, 794980, 0)  
INSERT INTO SessionLogFact VALUES (426, 1, 3, 0, 50400, 50400, 1970, 48430, 0, 0, 794980, 0)  
INSERT INTO SessionLogFact VALUES (427, 1, 4, 0, 50400, 50400, 1970, 48430, 48430, 0, 843410, 0)  
INSERT INTO SessionLogFact VALUES (428, 1, 1, 0, 50400, 50400, 1970, 48430, 200, 0, 843610, 0)  
INSERT INTO SessionLogFact VALUES (429, 1, 7, 170, 50400, 50570, 1970, 48600, 0, 0, 843610, 0)  
INSERT INTO SessionLogFact VALUES (430, 1, 6, 0, 50400, 50400, 1970, 48430, 0, 0, 843610, 0)  
INSERT INTO SessionLogFact VALUES (431, 1, 7, 170, 50400, 50570, 1970, 48600, 0, 0, 843610, 0)  
INSERT INTO SessionLogFact VALUES (432, 1, 6, 0, 50400, 50400, 1970, 48430, 0, 0, 843610, 0)  
INSERT INTO SessionLogFact VALUES (433, 1, 1, 0, 50400, 50400, 1970, 48430, 100, 0, 843710, 0)  
INSERT INTO SessionLogFact VALUES (434, 1, 1, 0, 50400, 50400, 1970, 48430, 0, 8000, 835710, 0)  
INSERT INTO SessionLogFact VALUES (435, 1, 3, 0, 50400, 50400, 1970, 48430, 0, 0, 835710, 0)  
INSERT INTO SessionLogFact VALUES (436, 1, 3, 0, 50400, 50400, 1970, 48430, 0, 0, 835710, 0)  
INSERT INTO SessionLogFact VALUES (437, 1, 5, 0, 50400, 50400, 2010, 48390, 1000, 0, 836710, 0)  
INSERT INTO SessionLogFact VALUES (438, 1, 2, 0, 50400, 50400, 2010, 48390, 0, 150, 836560, 0)  
INSERT INTO SessionLogFact VALUES (439, 1, 1, 0, 50400, 50400, 2010, 48390, 0, 0, 836560, 0)  
INSERT INTO SessionLogFact VALUES (440, 1, 5, 0, 50400, 50400, 2050, 48350, 1000, 0, 837560, 0)  
INSERT INTO SessionLogFact VALUES (441, 1, 4, 0, 50400, 50400, 2050, 48350, 48350, 0, 885910, 0)  
INSERT INTO SessionLogFact VALUES (442, 1, 3, 0, 50400, 50400, 2050, 48350, 10000, 0, 895910, 0)

INSERT INTO SessionLogFact VALUES (443, 1, 2, 0, 50400, 50400, 2050, 48350, 0, 110, 895800, 0)  
INSERT INTO SessionLogFact VALUES (444, 1, 5, 0, 50400, 50400, 2090, 48310, 1000, 0, 896800, 0)  
INSERT INTO SessionLogFact VALUES (445, 1, 4, 0, 50400, 50400, 2090, 48310, 48310, 0, 945110, 0)  
INSERT INTO SessionLogFact VALUES (446, 1, 1, 0, 50400, 50400, 2090, 48310, 100, 0, 945210, 0)  
INSERT INTO SessionLogFact VALUES (447, 1, 5, 0, 50400, 50400, 2130, 48270, 0, 0, 945210, 0)  
INSERT INTO SessionLogFact VALUES (448, 1, 7, 170, 50400, 50570, 2130, 48440, 0, 0, 945210, 0)  
INSERT INTO SessionLogFact VALUES (449, 1, 4, 170, 50400, 50570, 2130, 48440, 48440, 0, 993650, 0)  
INSERT INTO SessionLogFact VALUES (450, 1, 4, 170, 50400, 50570, 2130, 48440, 48440, 0, 1042090, 0)  
INSERT INTO SessionLogFact VALUES (451, 1, 6, 0, 50400, 50400, 2130, 48270, 0, 0, 1042090, 0)  
INSERT INTO SessionLogFact VALUES (452, 1, 7, 170, 50400, 50570, 2130, 48440, 0, 0, 1042090, 0)  
INSERT INTO SessionLogFact VALUES (453, 1, 6, 0, 50400, 50400, 2130, 48270, 0, 0, 1042090, 0)  
INSERT INTO SessionLogFact VALUES (454, 1, 4, 0, 50400, 50400, 2130, 48270, 48270, 0, 1090360, 0)  
INSERT INTO SessionLogFact VALUES (455, 1, 2, 0, 50400, 50400, 2130, 48270, 0, 30, 1090330, 0)  
INSERT INTO SessionLogFact VALUES (456, 1, 4, 0, 50400, 50400, 2130, 48270, 48270, 0, 1138600, 0)  
INSERT INTO SessionLogFact VALUES (457, 1, 3, 0, 51200, 51200, 2130, 49070, 0, 0, 1138600, 0)  
INSERT INTO SessionLogFact VALUES (458, 1, 7, 170, 51200, 51370, 2130, 49240, 0, 0, 1138600, 0)  
INSERT INTO SessionLogFact VALUES (459, 1, 1, 170, 51200, 51370, 2130, 49240, 150, 0, 1138750, 0)  
INSERT INTO SessionLogFact VALUES (460, 1, 5, 170, 51200, 51370, 2170, 49200, 1000, 0, 1139750, 0)  
INSERT INTO SessionLogFact VALUES (461, 1, 6, 0, 51200, 51200, 2170, 49030, 0, 0, 1139750, 0)  
INSERT INTO SessionLogFact VALUES (462, 1, 2, 0, 51200, 51200, 2170, 49030, 0, 120, 1139630, 0)  
INSERT INTO SessionLogFact VALUES (463, 1, 3, 0, 51200, 51200, 2170, 49030, 0, 0, 1139630, 0)  
INSERT INTO SessionLogFact VALUES (464, 1, 2, 0, 51200, 51200, 2170, 49030, 0, 200, 1139430, 0)  
INSERT INTO SessionLogFact VALUES (465, 1, 1, 0, 51200, 51200, 2170, 49030, 0, 0, 1139430, 0)  
INSERT INTO SessionLogFact VALUES (466, 1, 1, 0, 57200, 57200, 2170, 55030, 0, 80000, 1059430, 0)  
INSERT INTO SessionLogFact VALUES (467, 1, 3, 0, 57200, 57200, 2170, 55030, 0, 0, 1059430, 0)  
INSERT INTO SessionLogFact VALUES (468, 1, 3, 0, 57200, 57200, 2170, 55030, 0, 0, 1059430, 0)  
INSERT INTO SessionLogFact VALUES (469, 1, 4, 0, 57200, 57200, 2170, 55030, 55030, 0, 1114460, 0)  
INSERT INTO SessionLogFact VALUES (470, 1, 1, 0, 67200, 67200, 2170, 65030, 0, 150000, 964460, 0)  
INSERT INTO SessionLogFact VALUES (471, 1, 3, 0, 67200, 67200, 2170, 65030, 0, 0, 964460, 0)  
INSERT INTO SessionLogFact VALUES (472, 1, 1, 0, 67600, 67600, 2170, 65430, 0, 80000, 884460, 0)  
INSERT INTO SessionLogFact VALUES (473, 1, 7, 170, 67600, 67770, 2170, 65600, 0, 0, 884460, 0)  
INSERT INTO SessionLogFact VALUES (474, 1, 6, 0, 67600, 67600, 2170, 65430, 0, 0, 884460, 0)  
INSERT INTO SessionLogFact VALUES (475, 1, 1, 0, 97600, 97600, 2170, 95430, 0, 750000, 134460, 0)  
INSERT INTO SessionLogFact VALUES (476, 1, 1, 0, 97600, 97600, 2170, 95430, 0, 0, 134460, 0)  
INSERT INTO SessionLogFact VALUES (477, 1, 3, 0, 97600, 97600, 2170, 95430, 50000, 0, 184460, 0)  
INSERT INTO SessionLogFact VALUES (478, 1, 3, 0, 97600, 97600, 2170, 95430, 185000, 0, 369460, 0)  
INSERT INTO SessionLogFact VALUES (479, 1, 5, 0, 97600, 97600, 2210, 95390, 1000, 0, 370460, 0)  
INSERT INTO SessionLogFact VALUES (480, 1, 2, 0, 97600, 97600, 2210, 95390, 0, 20, 370440, 0)  
INSERT INTO SessionLogFact VALUES (481, 1, 1, 0, 97600, 97600, 2210, 95390, 0, 105000, 265440, 0)  
INSERT INTO SessionLogFact VALUES (482, 1, 5, 0, 97600, 97600, 2250, 95350, 1000, 0, 266440, 0)

INSERT INTO SessionLogFact VALUES (483, 1, 4, 0, 97600, 97600, 2250, 95350, 95350, 0, 361790, 0)  
INSERT INTO SessionLogFact VALUES (484, 1, 3, 0, 97600, 97600, 2250, 95350, 230000, 0, 591790, 0)  
INSERT INTO SessionLogFact VALUES (485, 1, 1, 0, 105600, 105600, 2250, 103350, 0, 220000, 371790, 0)  
INSERT INTO SessionLogFact VALUES (486, 1, 5, 0, 105600, 105600, 2290, 103310, 1000, 0, 372790, 0)  
INSERT INTO SessionLogFact VALUES (487, 1, 4, 0, 105600, 105600, 2290, 103310, 103310, 0, 476100, 0)  
INSERT INTO SessionLogFact VALUES (488, 1, 1, 0, 110600, 110600, 2290, 108310, 0, 150000, 326100, 0)  
INSERT INTO SessionLogFact VALUES (489, 1, 5, 0, 110600, 110600, 2330, 108270, 1000, 0, 327100, 0)  
INSERT INTO SessionLogFact VALUES (490, 1, 7, 170, 110600, 110770, 2330, 108440, 0, 0, 327100, 0)  
INSERT INTO SessionLogFact VALUES (491, 1, 4, 170, 110600, 110770, 2330, 108440, 108440, 0, 435540, 0)  
INSERT INTO SessionLogFact VALUES (492, 1, 4, 170, 110600, 110770, 2330, 108440, 108440, 0, 543980, 0)  
INSERT INTO SessionLogFact VALUES (493, 1, 3, 170, 110600, 110770, 2330, 108440, 0, 0, 543980, 0)  
INSERT INTO SessionLogFact VALUES (494, 1, 1, 170, 110600, 110770, 2330, 108440, 0, 0, 543980, 0)  
INSERT INTO SessionLogFact VALUES (495, 1, 6, 0, 110600, 110600, 2330, 108270, 0, 0, 543980, 0)  
INSERT INTO SessionLogFact VALUES (496, 1, 4, 0, 110600, 110600, 2330, 108270, 108270, 0, 652250, 0)  
INSERT INTO SessionLogFact VALUES (497, 1, 2, 0, 110600, 110600, 2330, 108270, 0, 200, 652050, 0)  
INSERT INTO SessionLogFact VALUES (498, 1, 4, 0, 110600, 110600, 2330, 108270, 108270, 0, 760320, 0)  
INSERT INTO SessionLogFact VALUES (499, 1, 3, 0, 110600, 110600, 2330, 108270, 0, 0, 760320, 0)  
INSERT INTO SessionLogFact VALUES (500, 1, 7, 170, 110600, 110770, 2330, 108440, 0, 0, 760320, 0)  
INSERT INTO SessionLogFact VALUES (501, 1, 1, 170, 122600, 122770, 2330, 120440, 0, 350000, 410320, 0)  
INSERT INTO SessionLogFact VALUES (502, 1, 5, 170, 122600, 122770, 2370, 120400, 1000, 0, 411320, 0)  
INSERT INTO SessionLogFact VALUES (503, 1, 3, 170, 122600, 122770, 2370, 120400, 0, 0, 411320, 0)  
INSERT INTO SessionLogFact VALUES (504, 1, 2, 170, 122600, 122770, 2370, 120400, 0, 800, 410520, 0)  
INSERT INTO SessionLogFact VALUES (505, 1, 6, 0, 122600, 122600, 2370, 120230, 0, 0, 410520, 0)  
INSERT INTO SessionLogFact VALUES (506, 1, 2, 0, 122600, 122600, 2370, 120230, 0, 90, 410430, 0)  
INSERT INTO SessionLogFact VALUES (507, 1, 1, 0, 122600, 122600, 2370, 120230, 0, 0, 410430, 0)  
INSERT INTO SessionLogFact VALUES (508, 1, 1, 0, 123300, 123300, 2370, 120930, 0, 110000, 300430, 0)  
INSERT INTO SessionLogFact VALUES (509, 1, 3, 0, 123800, 123800, 2370, 121430, 0, 0, 300430, 0)  
INSERT INTO SessionLogFact VALUES (510, 1, 7, 170, 123800, 123970, 2370, 121600, 0, 0, 300430, 0)  
INSERT INTO SessionLogFact VALUES (511, 1, 1, 170, 124000, 124170, 2370, 121800, 0, 800, 299630, 0)  
INSERT INTO SessionLogFact VALUES (512, 1, 5, 170, 124000, 124170, 2410, 121760, 1000, 0, 300630, 0)  
INSERT INTO SessionLogFact VALUES (513, 1, 3, 170, 124000, 124170, 2410, 121760, 0, 0, 300630, 0)  
INSERT INTO SessionLogFact VALUES (514, 1, 2, 170, 124000, 124170, 2410, 121760, 0, 70, 300560, 0)  
INSERT INTO SessionLogFact VALUES (515, 1, 6, 0, 124000, 124000, 2410, 121590, 0, 0, 300560, 0)  
INSERT INTO SessionLogFact VALUES (516, 1, 2, 0, 124000, 124000, 2410, 121590, 0, 100, 300460, 0)  
INSERT INTO SessionLogFact VALUES (517, 1, 1, 0, 126000, 126000, 2410, 123590, 0, 30000, 270460, 0)  
INSERT INTO SessionLogFact VALUES (518, 1, 1, 0, 134000, 134000, 2410, 131590, 0, 80000, 190460, 0)  
INSERT INTO SessionLogFact VALUES (519, 1, 3, 0, 134000, 134000, 2410, 131590, 0, 0, 190460, 0)  
INSERT INTO SessionLogFact VALUES (520, 1, 3, 0, 134000, 134000, 2410, 131590, 0, 0, 190460, 0)  
INSERT INTO SessionLogFact VALUES (521, 1, 4, 0, 134000, 134000, 2410, 131590, 131590, 0, 322050, 0)  
INSERT INTO SessionLogFact VALUES (522, 1, 1, 0, 134700, 134700, 2410, 132290, 0, 170000, 152050, 0)

INSERT INTO SessionLogFact VALUES (523, 1, 7, 170, 134700, 134870, 2410, 132460, 0, 0, 152050, 0)  
INSERT INTO SessionLogFact VALUES (524, 1, 6, 0, 134700, 134700, 2410, 132290, 0, 0, 152050, 0)  
INSERT INTO SessionLogFact VALUES (525, 1, 7, 170, 134700, 134870, 2410, 132460, 0, 0, 152050, 0)  
INSERT INTO SessionLogFact VALUES (526, 1, 6, 0, 134700, 134700, 2410, 132290, 0, 0, 152050, 0)  
INSERT INTO SessionLogFact VALUES (527, 1, 1, 0, 134700, 134700, 2410, 132290, 0, 0, 152050, 0)  
INSERT INTO SessionLogFact VALUES (528, 1, 1, 0, 134700, 134700, 2410, 132290, 0, 70000, 82050, 0)  
INSERT INTO SessionLogFact VALUES (529, 1, 3, 0, 134700, 134700, 2410, 132290, 230000, 0, 312050, 0)  
INSERT INTO SessionLogFact VALUES (530, 1, 3, 0, 134700, 134700, 2410, 132290, 0, 0, 312050, 0)  
INSERT INTO SessionLogFact VALUES (531, 1, 3, 0, 134700, 134700, 2410, 132290, 0, 0, 312050, 0)  
INSERT INTO SessionLogFact VALUES (532, 1, 3, 0, 134700, 134700, 2410, 132290, 3000, 0, 315050, 0)  
INSERT INTO SessionLogFact VALUES (533, 1, 4, 0, 134700, 134700, 2410, 132290, 132290, 0, 447340, 0)  
INSERT INTO SessionLogFact VALUES (534, 1, 1, 0, 137700, 137700, 2410, 135290, 0, 60000, 387340, 0)  
INSERT INTO SessionLogFact VALUES (535, 1, 7, 170, 137700, 137870, 2410, 135460, 0, 0, 387340, 0)  
INSERT INTO SessionLogFact VALUES (536, 1, 6, 0, 137700, 137700, 2410, 135290, 0, 0, 387340, 0)  
INSERT INTO SessionLogFact VALUES (537, 1, 7, 170, 137700, 137870, 2410, 135460, 0, 0, 387340, 0)  
INSERT INTO SessionLogFact VALUES (538, 1, 6, 0, 137700, 137700, 2410, 135290, 0, 0, 387340, 0)  
INSERT INTO SessionLogFact VALUES (539, 1, 1, 0, 142700, 142700, 2410, 140290, 0, 200000, 187340, 0)  
INSERT INTO SessionLogFact VALUES (540, 1, 1, 0, 142700, 142700, 2410, 140290, 0, 0, 187340, 0)  
INSERT INTO SessionLogFact VALUES (541, 1, 3, 0, 142700, 142700, 2410, 140290, 0, 0, 187340, 0)  
INSERT INTO SessionLogFact VALUES (542, 1, 7, 170, 142700, 142870, 2410, 140460, 0, 0, 187340, 0)