

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ АВТОМАТИЗАЦІЇ
КЛІРІНГУ У МІЖНАРОДНІЙ ПЛАТІЖНІЙ СИСТЕМІ UNIONPAY
SOFTWARE MODULE DEVELOPMENT FOR CLEARING
AUTOMATISATION IN THE UNIONPAY INTERNATIONAL
PAYMENT SYSTEM

Виконав: студент(ка) 4 курсу, групи КНТз-118
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Косінов А.В.

(прізвище та ініціали)

Керівник Субботін С.О.

(прізвище та ініціали)

Рецензент Гофман Є.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Косінова Андрія Вікторовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного модуля для автоматизації клірингу у міжнародній платіжній системі UnionPay. Software module development for clearing automatisisation in the UnionPay international payment system.

керівник проєкту (роботи) Субботін Сергій Олександрович, д.т.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Субботін С.О., зав. каф.		
Нормоконтроль	Камінська Ж.К., асистент		

7. Дата видачі завдання «04» квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Срок виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тижні	Розділ 2
4	Розробка архітектури програми.	2 тижні	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Косінов А.В.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Субботін С.О.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до випускної кваліфікаційної роботи бакалавра:
71 с., 31 рис., 13 джерел.

DATABASE MODULE, ORACLE, TOAD FOR ORACLE, SQL
LOADER, SQL\PLSQL.

Об'єкт дослідження – процес розробки та функціонування модуля БД
для роботи з кліринговою системою.

Предмет дослідження – методи розробки та підтримки функціонування
модуля БД призначеного для обробки транзакцій клірингової системи.

Мета роботи – розробити модуль БД який автоматизує кліринг у
міжнародній системі UPI.

Методи дослідження – методи збору та аналізу вимог до програмного
забезпечення, методи моделювання, конструювання та тестування програмного
забезпечення.

При розробці модуля БД було проведено аналіз предметної області,
спроектована функціональність, реалізовано модуль за допомогою
інструментарію Toad for Oracle, мови програмування SQL\PLSQL та бази
даних Oracle, здійснене тестування модулю БД. В результаті роботи отримано
модуль БД, який автоматизує кліринг у системі UPI.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 71 pages, 31 figures, 13 references.

DB MODULE, ORACLE, TOAD FOR ORACLE, SQL LOADER, SQL\PLSQL.

The object of research is the process of development and operation of the database module for working with the clearing system.

The subject of research are methods of development and maintenance of the database module designed for processing transaction in clearing system.

The purpose of the work is to develop a database module that automates clearing in the UPI international system.

Research method - methods of collecting and analyzing software requirements, methods of modeling, designing and testing software.

During the development of the database module, the subject was analyzed, the functionality was designed, the database module was implemented using the Toad for Oracle tool, the SQL\PLSQL programming language and the Oracle database, the database module was tested. As a result a database module was obtained that automates payments in the UPI clearing system.

ЗМІСТ

С.

Перелік скорочень та умовних позначок.....	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Мета розробки модуля БД.....	9
1.2 БД Oracle	10
2 Проектування модулю БД	19
2.1 Огляд використовуваних інструментів	19
2.1.1 SQL Loader.....	19
2.1.2 Toad for Oracle	20
2.1.3 Мова запитів SQL	22
2.1.4 Мова програмування PL\SQL.....	23
2.2 Архітектура.....	25
3 Реалізація модулю БД.....	30
3.1 Опис функціональності.....	30
3.2 Розробка модулю БД.....	30
3.2.1 Вхідні файли для обробки з UPI.....	30
3.2.2 Розробка таблиць БД.....	33
3.2.3 Розробка контролів файлів для завантаження даних.....	34
3.2.4 Розробка специфікації пакета БД.....	35
3.2.5 Розробка процедур БД.....	37
3.2.6 Розробка функцій БД	38
4 Тестування	39
Висновки	43
Перелік посилань.....	44
Додаток А Код створення таблиць	46
Додаток Б Код керуючих файлів.....	50
Додаток В Код специфікації пакету БД.....	54
Додаток Г Код тілу пакету БД.....	56
Додаток Д Слайди презентації.....	61

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ACID	– Atomicity, Consistency, Isolation, Durability;
DDL	– Data Definition Language;
DCL	– Data Control Language;
DML	– Data Manipulation Language;
ELT	– Extract, Load, Transform;
IRF	– Interchange reimbursement fee;
JVM	– Java Virtual Machine;
PGA	– Process-Program Global Area;
SGA	– System Global Area;
TCL	– Transaction Control Language;
UGA	– User Global Area;
UPI	– UnionPay International;
БД	– база даних;
СУБД	– система управління базами даних.

ВСТУП

Важко уявити в сучасній світовій економіці компанію, підприємство, банк чи країну яка не користується цифровими, технологічними платіжними системами. Visa, Mastercard чи UPI, на мою думку, дуже міцно та на довгий час зафіксувались як постійні та незмінні помічники, які дозволяють виконувати різноманітні безготівкові фінансові операції. З появою та удосконаленням платіжних систем виріс рівень захисту заощаджень, простота застосування при сплаті, швидкість проведення розрахунків. Ці інновації дуже сильно вплинули та змінили життя мільйонів людей.

В даний час платіжна система UPI знаходиться на піку свого розвитку та широко впроваджується у всіх економічно розвинутих країнах. З появою цієї платіжної системи активно розвиваються і впроваджуються нові пристрої та платформи на ряду з цим стрімко розвивається сфера розробки платежів в додатках та розробки БД.

Метою дипломної кваліфікаційної роботи є розробка модуля БД для автоматизації клірингу в платіжній системі UPI. Потреба виникла у зв'язку з тим, що на українському ринку здебільшого використовуються платіжні системи Visa та Mastercard – цей фінансовий ринок вже доволі насичений. Китайська система UPI нині займає у світі лідируючі позиції, проте на ринку України система UPI ще мало розповсюджена, тому має великі можливості та перспективи.

Дипломна робота складається з вступу, чотирьох розділів, висновків, літератури та додатків.

Перший розділ містить опис мети розробки, також були розглянуті існуючі БД.

Другий розділ містить опис засобів розробки модулю БД.

Третій розділ складається з процесу розробки модулю БД.

Четвертий розділ включає в себе тестування створеного модулю.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Мета розробки модуля БД

В наш час міжнародні карткові системи такі як: Visa та Mastercard - дуже популярні. Люди з усього світу використовують картки для безготівкових розрахунків, оплати різних послуг, товарів, використання кредитів та депозитів. В даний час платіжна система UPI контролює сорок п'ять відсотків світового ринку всіх платіжних карток. Україна стала сто сімдесят восьмою країною, в якій буде представлена китайська компанія. Для банків України необхідно розширити свої функціональні можливості для розширення послуг населенню, роблячи життя людей більш комфортним і якісним.

Ідея створення цієї роботи виникла з метою забезпечення для фінансової компанії програмним модулем, який автоматизує кліринг в міжнародній системі UPI. Компанії які можуть бути потенційними клієнтами модуля який розробляється, як правило не мають професійного досвіду проектування і розробки подібних систем так, як ринок для UPI тільки починає розвиватися. Розробка таких систем силами компанії вимагає капітальних вкладень у людські та матеріальні ресурси, апаратні та програмні компоненти, що може привести до втрати часу, коштів та іміджу компаній.

Щоб уникнути вищевказаних наслідків було прийнято рішення створити програмний модуль для автоматизації клірингу в китайській системі UPI. Для роботи програмного модуля використовується реляційна БД в якій дані атомарні, узгоджені, ізольовані та довговічні. У модулі передбачено механізм ELT – вилучення, завантаження та перетворення даних до зручного формату та подальшої обробки. Можливе сполучення модуля з такими системами як Visa, Mastercard.

Використання БД надає перспективи для розширення функціональних можливостей платіжної системи UPI, додаючи необхідні додаткові модулі.

З часом, залежно від вимог ринку та потреб компаній, можна використовувати такі сервіси як: Web інтерфейс, інтерфейс мобільних додатків, різні бібліотеки та мови програмування (java, python та інші).

1.2 БД ORACLE

Oracle Database є об'єктно-реляційною СУБД від компанії Oracle, перший випуск якої відбувся у 1979 році, була написана на мові програмування Assembler та працювала під управлінням ОС RSX-11. На даний час СУБД має безліч різних версій та типів, які були розроблені на мовах програмування C та C++ та працюють під різними ОС такі, як:

- а) Windows x86-64;
- б) Linux x86-64;
- в) Solaris x86-64;
- г) IBM;
- д) Mac OS x Server.

Вище вказано що СУБД Oracle є об'єктно-реляційною, тобто модель зберігання даних поєднує в собі об'єктно-орієнтовану і реляційну моделі. Реляційна модель представляє із себе набір відносин між записами які формують БД, її можна уявити як двомірну таблицю зі стовпами та строками, приклад наведено на рисунку 1.1.

Для того щоб зв'язати дані які знаходяться в різних таблицях використовують унікальний ідентифікатор строки. Реальні проекти можуть вміщувати тисячі таблиць, а кожна таблиця мільйони записів, внаслідок цього читання інформації вручну майже неможливе, тому для роботи с БД використовується мова структурованих запитів SQL, яка буде розглядатись далі.

The screenshot shows the Toad for Oracle interface. The 'Schema Browser' pane on the left displays the database structure, including tables like UPL_AC, UPL_AC_TEMP, UPL_AC_TEMP, UPL_AFEEDIT, UPL_BN_HEADER, and UPL_BN_REC. The main pane shows the 'UPL_AC_TEMP' table with its columns and data. The table has 210 rows, with the first row being a header row and the subsequent rows containing numerical data.

ACQUIRING_ICC	FORWARDING_ICC	SYSTEM_TRACE_AUDIT	TRANSMISSION_DATE	PAN	AMOUNT_TRANSACTION	MESSAGE_TYPE	PROCESSING_CODE	MERCHANT_TYPE	CARD_ACCEPTOR_TERMINAL_ID	CARD_ACCEPTOR_ICC	CARD_ACCEPTOR_NAME_LOCATION
44430804	00049992	357311	0901121815	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	385188	0901131626	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	882186	0902132519	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	114276	1129190226	6250947000000014	000000001000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	487151	1130161656	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	953651	1123210516	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	976649	1123219515	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	517149	1124215958	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	535276	1127061400	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	832901	1201172752	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	794526	1201155626	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	152649	1208182732	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	586277	1208182702	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	972401	1213172237	6250947000000014	0000000001500	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	474777	1213172754	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	431275	1213172555	6250947000000014	000000000100	0220	200000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	513652	121303108	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	473652	1213172511	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	590399	0208020811	6250947000000014	0000000002300	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	534401	02080213629	6250940500000006	0000000000500	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	501435	0825121143	6250940500000006	000000010000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	482687	0825115859	6250947000000014	000000010000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	560935	0825160518	6250947000000014	000000010000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	755186	0825120436	6250947000000014	000000009000	0200	000000	5311	01080209	3810000500000001	3810000500000001
44430804	00049992	504517	0825174610	6250947000000014	000000010000	0200	000000	5311	01080209	3810000500000001	3810000500000001

Рисунок 1.1 – Приклад реляційної БД

Об'єктно-орієнтована модель сприймає дані як об'єкт: тобто таблиця БД є - об'єктом, поля таблиці - її властивостями, а одна строчка (транзакція) - як екземпляр таблиці. Цей підхід можна порівняти з ООП у якому об'єкт є екземпляром деякого класу який має певні властивості та методи.

По структурі зберігання даних є інша модель –нереляційна. Такі БД зберігають дані у вигляді древа - приклад наведено на рисунку 1.2а, графа, у форматі ключ-значення - приклад наведено на рисунку 1.2 б нижче.

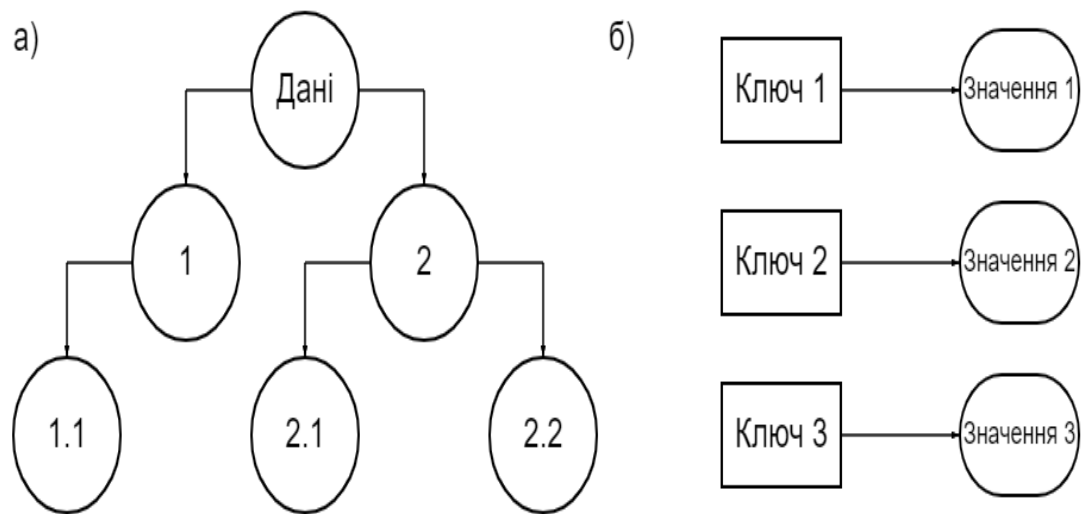


Рисунок 1.2 – Структури зберігання даних в нереляційних БД

Тобто БД це структура зберігання, зміни та взаємодії з інформацією. Але нам потрібно взаємодіяти з інформацією, для цього ми використовуємо СУБД, яка підтримує мову запитів SQL і дозволяє нам:

- а) налаштовувати безпеку та конфіденційність інформації;
- б) роботи запити у вигляді коду для знаходження та отримання даних;
- в) створювати резервні копії інформації та відновлювати базу після збоїв;
- г) створювати нові записи, редагувати їх та видаляти;
- д) об'єднувати дані які зберігаються у різних місцях, також видаляти їх;
- е) виконувати паралельно різні дії і керувати доступом к інформації;
- є) керувати транзакціями.

СУБД Oracle була спроектована як переносна, доступна на усіх великих платформах від UNIX/Linux і Windows до мейнфреймів. Однак фізична архітектура в Oracle в різних ОС відрізняється. Наприклад в UNIX/Linux СУБД Oracle реалізована як сукупність процесів ОС, по одному процесу на кожну тяжку функцію. Для UNIX/Linux систем така реалізація цілком природна, оскільки робота відбувається в середовищі з безліччю процесів. Проте, в Windows ця архітектура була б невідповідною та працювала б не особливо ефективно (система була б повільною і не піддавалась масштабуванню). Тому на платформі Windows СУБД Oracle реалізована в вигляді одного процесу з безліччю потоків. Незважаючи на те, що фізичні механізми, що застосовуються в реалізаціях Oracle варіюються від платформи до платформи, архітектура – є достатньою мірою узагальненою, для того щоб можливо було отримати гарне уявлення про її роботу на всіх платформах [1].

Існують терміни при використанні в контексті Oracle, які викликають плутанину – екземпляр та БД. База даних – колекція фізичних файлів чи дисків ОС. В дипломній роботі використовується версія Oracle Database 12.2.0.1.0 в даній версії використовується три типи бази даних, розглянемо їх нижче:

а) БД з єдиним власником. Це відокремлений набір файлів даних, керуючих файлів, журнальних файлів для відновлення, файлів параметрів, який також до всіх метаданих, даних і коду програми містить всі метадані Oracle. У версіях що передували 12, це був єдиний тип бази даних;

б) контейнерна або коренева БД. Це відокремлений набір файлів даних, керуючих файлів, журнальних файлів для відновлення, файлів параметрів і т.д., який включає тільки метадані Oracle, дані Oracle та код Oracle. У таких файлах даних відсутні об'єкти або код програми - тільки метадані та об'єкти коду, надані Oracle. БД є відокремленою тому, що вона може бути змонтована та відкрита без додаткових підтримуючих фізичних структур;

в) база даних, що підключається (pluggable). Це набір лише файлів даних. Така база даних не є відокремленою. Щоб з'явилася можливість відкриття та доступу, БД, що підключається, повинна бути "вставлена" в

контейнерну базу даних. Файли даних містять тільки метадані для об'єктів програм, дані програм та код для програм. Метадані або будь-який код Oracle у цих файлах даних відсутній. Немає жодних журнальних файлів для відновлення, керуючих файлів, файлів параметрів тощо. - тільки файли даних, асоційовані з базою, що підключається даних. БД, що підключається, успадковує інші типи файлів від контейнерної бази даних, до якої вона підключена в поточний момент.

Екземпляр (Instance). Набір фонових процесів (Background processes) або потоків і області пам'яті, що розділяється Oracle, що представляє собою пам'ять, яка спільно використовується цими потоками або процесами, що виконуються на одному комп'ютері. Це місце для мінливих, непостійних даних, частина яких скидається на диск. Примірник БД (Database) може існувати взагалі без дискового сховища. Системна глобальна область (SGA) та процеси, рисунок 1.3 [2].

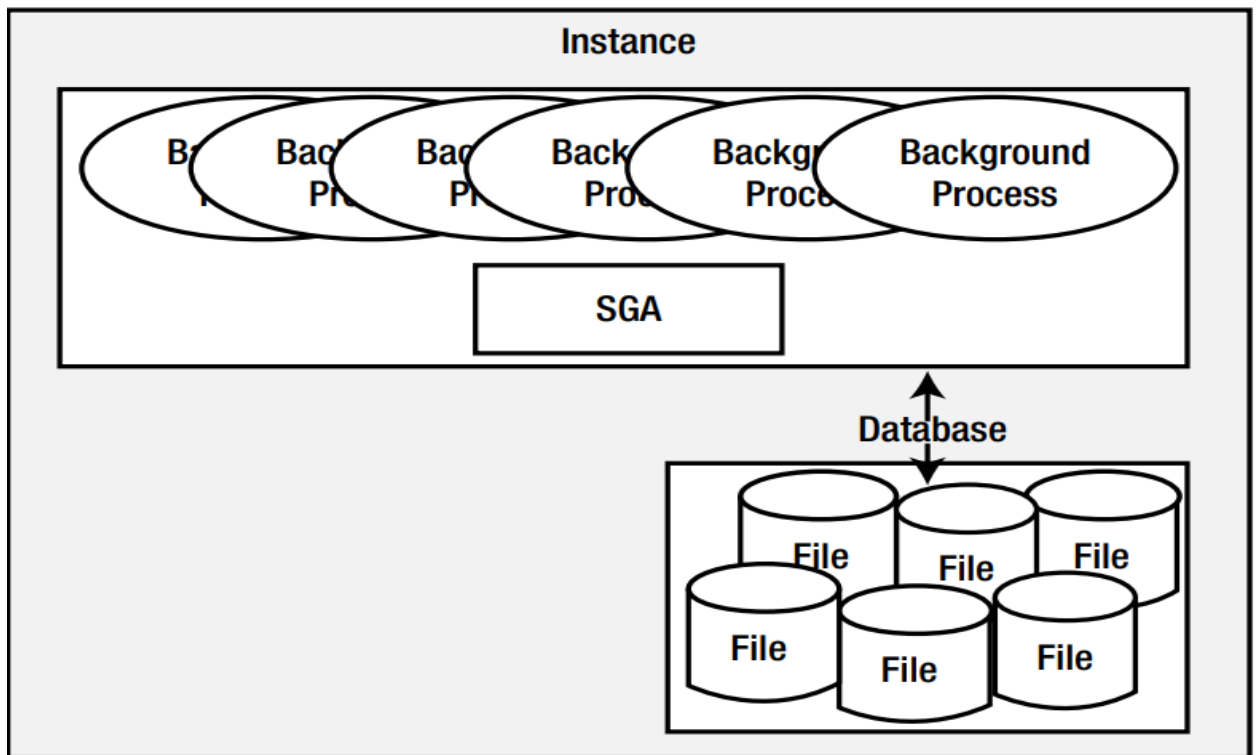


Рисунок 1.3 – Схема екземпляру БД в Oracle [2]

Також необхідно пригадати що у Oracle є три основні структури пам'яті:

а) глобальна область системи – SGA. Великий спільно використовуваний сегмент пам'яті, в який в той чи інший момент часу будуть звертатися практично всі процеси Oracle;

б) глобальна область процесу чи програми – PGA. Закрита область окремого процесу чи потоку, з інших процесів чи потоків вона не доступна;

в) глобальна область користувача – UGA. Ця область пам'яті пов'язана із конкретним сеансом. Вона розташована або в області SGA, якщо підключення до БД виконано за допомогою сервера, що розділяється, або в області PGA, якщо підключення до БД здійснено через виділений сервер.

SGA поділена на деякі пули: Java Pool – фіксований обсяг пам'яті, виділеної для віртуальної машини JVM, яка діє в базі даних; Large Pool - пул використовується підключеннями за допомогою сервера для пам'яті сеансу, засобами паралельного виконання для буферів повідомлень та резервним копіюванням (RMAN) для буферів дискового введення-виводу. Великий пул допускає оперативну зміну розміру; Shared Pool - пул містить курсори де зберігаються процедури, об'єкти стану, кеші словника даних і багато десятків інших елементів даних; Streams Pool - цей пул пам'яті використовується інструментами спільного використання даних, такими як Oracle GoldenGate, Oracle Streams і т.д. Якщо він не налаштований, але функціональність Streams задіяна, то Oracle виділить під пам'ять потоків до 10 відсотків обсягу пулу, що розділяється; Null Pool - являє собою пам'ять, призначену для зберігання буферів блоків (Block Buffers), кешованих блоків БД, буфера журналу повторення (Redo Buffer). Область SGA наведена нижче на рисунку 1.4 [3].

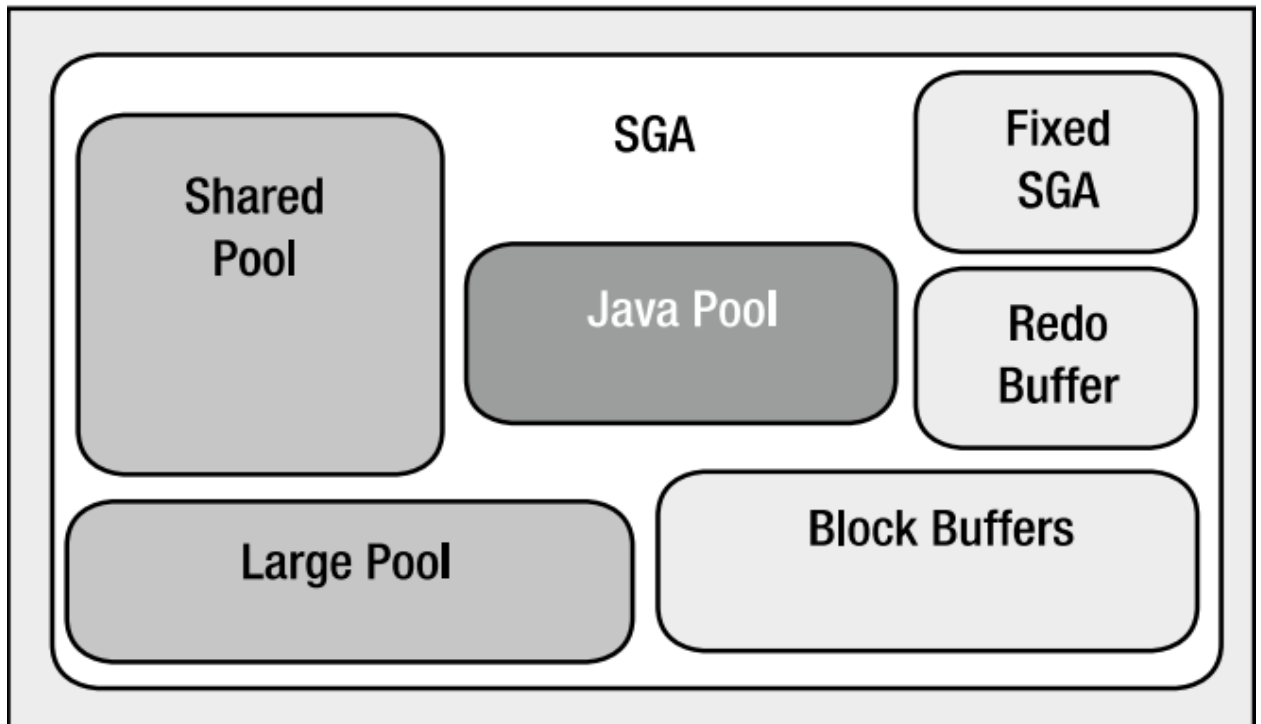


Рисунок 1.4 – Область SGA [3]

Область PGA - це "купа" пам'яті, у якій можуть розподілятися інші структури. Область UGA також є " купою " пам'яті, у якій можна визначити різні структури, притаманні сеансу. Область UGA розподіляється всередині області PGA при підключенні до бази даних Oracle за допомогою виділеного сервера та області SGA при підключенні через сервер, що розділяється.

З цього випливає, що при використанні сервера, що розділяється, необхідно визначати розмір великого пулу області SGA так, щоб забезпечити достатній обсяг для обслуговування будь-якого можливого користувача, який будь-коли підключиться до бази даних паралельно. Таким чином, область SGA бази даних, яка підтримує підключення за допомогою сервера, що розділяється, в загальному випадку набагато більше області SGA аналогічно конфігуровано бази даних, що підтримує тільки режим виділеного сервера.

Наступним необхідно розібрати транзакції, як засіб який відрізняє БД від файлових систем. Якщо ми знаходимось на середині запису файлу а ОС аварійно завершує свою роботу, то цей файл буде пошкодженим. Треба сказати, що є файлові системи з веденням журналів, які можуть відновити

файл на якийсь момент часу в минулому. Тим не менш, якщо потрібно утримувати два файли в синхронізованому стані, то такі системи в цьому не допоможуть, якщо оновити один файл, а в системі виникне відмова перш ніж буде завершено оновлення другого файлу, файли не будуть синхронізовані. В цьому і полягає основне призначення транзакцій, вони переводять БД з одного узгодженого стану до іншого – це і є їх головною функцією. Коли фіксується робота в БД, ми можемо бути впевненими, що будуть збережені або всі внесені зміни, або ні одне зі змін. Більш того, гарантується реалізація різноманітних правил і перевірок, що захищають цілісність даних. Транзакції в Oracle задовольняє усім характеристикам ACID:

- а) атомарність - виконуються або всі події, вироблені в транзакції, або жодна;
- б) узгодженість - транзакція переміщує базу даних із одного узгодженого стану до іншого;
- в) ізоляційність - результати транзакції можуть бути непомітні іншим транзакціям доти, доки вона не буде зафіксована;
- г) постійність - як тільки транзакція зафіксована, вона залишається постійною.

В Oracle немає потреби в операторі типу "розпочати транзакцію", вона починається неявно з першим оператором, який створює, видаляє або модифікує дані. Транзакцію можна розпочати явно з використанням оператора SET TRANSACTION або пакета DBMS_TRANSACTION, але в Oracle цей крок не обов'язковий на відміну від інших СУБД. Оператор COMMIT або ROLLBACK явно завершує транзакцію.

В заключенні цього розділу можна сказати, що Oracle входить в число багатокористувацьких систем яка удосконалювалась протягом великого проміжку часу. Дана СУБД є основним промисловим рішенням у галузі обробки даних. СУБД Oracle має найбільш розвинений функціонал для роботи з Java, на це також у великій кількості вплинуло те, що компанія Oracle поглинула компанію Sun Microsystems – розробника мови програмування Java.

Підбиваючи підсумки можна сказати що СУБД Oracle поєднала в собі велику кількість переваг:

а) має високий рівень надійності, фактично це зразок надійності серед подібних систем;

б) найсвіжіші інновації та вражаючий функціонал вже впроваджено у цьому продукті, оскільки компанія Oracle займає лідируюче місце та прагне тримати планку і далі;

в) має функціонал для розбиття великих БД на розділи, що надає можливість керувати багато-гігабайтними базами з високою ефективністю;

г) підтримуються великі обсяги пам'яті та симетрична багатопроцесорна обробка, що дозволяє керувати програмами при високих навантаженнях, пов'язаних із великою кількістю транзакцій;

д) підтримується на всіх відомих платформах.

Єдиний мінус цієї системи – її ціна. Але ця система підходить для великих організацій, які працюють з великим обсягом даних та різноманітними функціями. Такі компанії піклуються про якість надання своїх сервісів та послуг, тому готові витратити гроші на комерційний продукт.

2 ПРОЕКТУВАННЯ МОДУЛЮ БД

2.1 Огляд використовуваних інструментів

2.1.1 SQL*Loader

SQL Loader – утиліта яка завантажує дані із зовнішніх файлів у таблиці БД Oracle. Утиліта має потужний механізм синтаксичного аналізу даних у файлах даних, використовується для того щоб:

- а) завантажувати дані через мережу, якщо файли даних знаходяться в системі, яка відрізняється від БД;
- б) завантажувати дані з декількох файлів даних під час одного сеансу;
- в) завантажувати дані в декілька таблиць під час одного сеансу завантаження;
- г) маніпулювати даними до їх завантаження, використовуючи функції SQL;
- д) генерувати унікальні послідовні значення ключів у зазначених стовпцях;
- е) завантажувати доволі складні об'єктно-реляційні дані.

SQL Loader можливо використовувати двома способами: з файлом керування та без нього. Контрольний файл (Loader Control file) керує поведінкою утиліти та одного або декількох файлів даних (Input Datafiles), які використовуються під час завантаження. Контрольні файли надають більш ширшу варіативність над операцією завантаження для складних ситуацій, для простих завантажень використовується без контрольних файлів. Результатом є БД (Database) в яку завантажуються дані, файл журналу (Log File), файл з відхиленими записами (Bad Files), якщо такий є, та файл скидання (Discard Files), схему перетворення вхідних даних наведено на рисунку 2.1 [4].

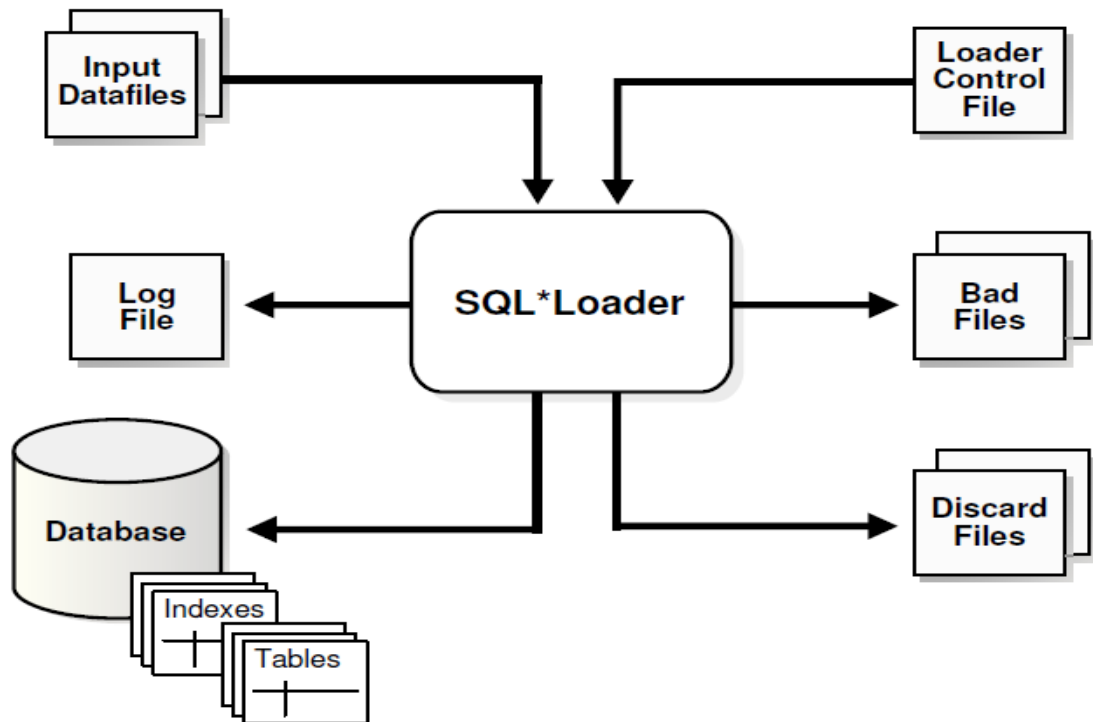


Рисунок 2.1 – Схема перетворення вхідних даних [4]

2.1.2 Tool for Oracle Application Developers

Toad – програмне забезпечення яке використовується для роботи з реляційними СУБД, розроблене компанією Dell. ПЗ повністю автоматизує задачі розробки і адміністрування програм на платформі Oracle. Для подальшого написання модуля було обрано програмне середовище Toad for Oracle тому, що має такі особливості:

- а) максимальна ефективність під час вирішення щоденних завдань адміністрування;
- б) максимальна точність при розробці, налагодженні та оптимізації коду;
- в) виконує різні операції розробки та адміністрування за допомогою одного інструменту;
- г) швидке перемикання між завданнями;
- д) розширена система підготовки звітів, що дозволяє публікувати велику кількість показників;

е) можливість доступу до додаткової експертизи роботи з Oracle через групу користувачів Toad (<http://toadfororacle.com/index.jspa>) [5].

ПЗ дозволяє швидко переміщуватися загальною ієрархією в пошуках потрібного об'єкта, вся інформація про обраний об'єкт відразу ж відображається в спеціальному вікні. Функція (Schema Browser) дає змогу швидко переглядати схему даних. Графічне середовище розробки підтримує одночасно з різні мови програмування (SQL, PL\SQL, HTML, JAVA). Графічний інтерфейс програмного середовища Toad for Oracle можемо побачити на рисунку 2.2.

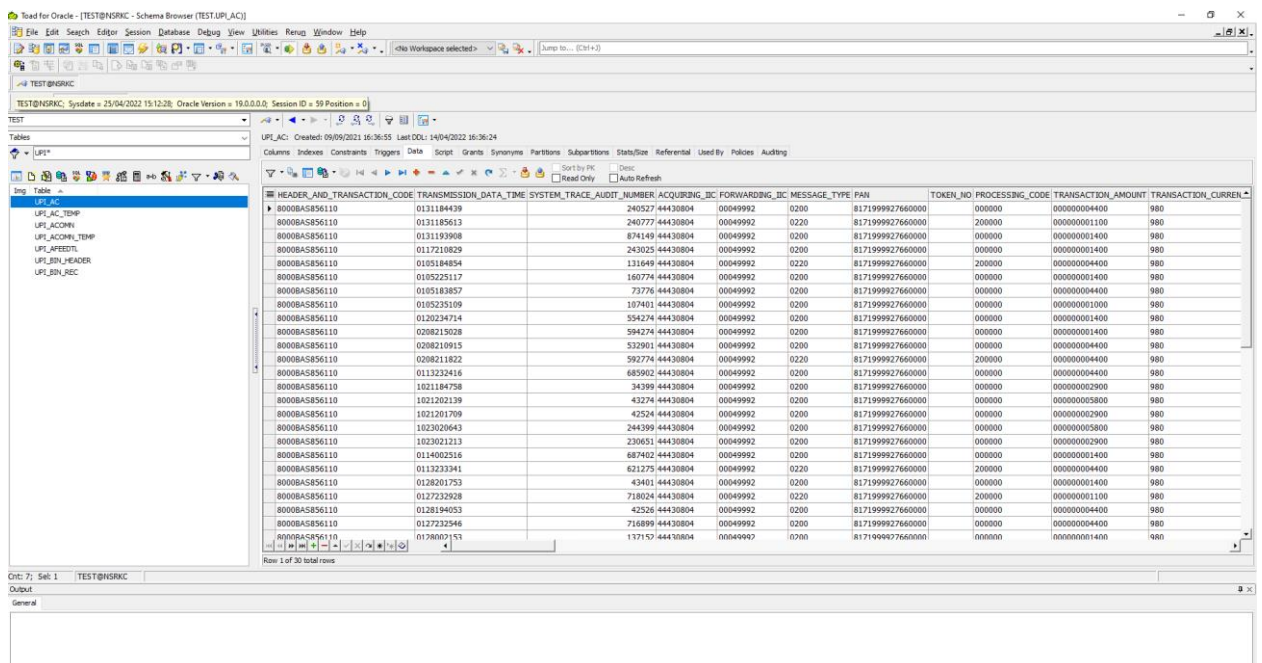


Рисунок 2.2 – Інтерфейс програми Toad for Oracle

До переваг також можна віднести редактор SQL Editor з режимом автоматичної перевірки та випереджальним введенням коду, має закладки для швидкого переходу в потрібну область коду; панель інструментів; одночасну компіляцію кількох об'єктів з узгодженою компіляцією всіх залежних об'єктів; поділ шаблонів, скриптів, налаштувань, ділянок коду з іншими учасниками проекту; запис етапів проекту та подальше відтворення. За допомогою інструменту швидкого створення запитів – SQL Modeler можна

задавати критерії запиту, створювати запити автоматично, перевіряти генеровані запити, також можливе створення запитів до БД на інших платформах. Відладчик дозволяє відслідковувати всі етапи роботи програми, переглядати будь яку кількість змінних і при необхідності змінювати їх значення, налагоджувати оператори по одному, також налагоджувати частини процедур. Модуль управління DBA поєднує середовище для створення БД та їх адміністрування використовуючи порівняння та синхронізацію схем БД, перегляд журналів, створення пакетних завдань та зміна параметрів. Серед комунікаційних можливостей імпорт та експорт даних, підтримка SQL Loader [6].

2.1.3 Мова запитів SQL

SQL (Structured Query Language) – структурована мова запитів яка дозволяє створювати, видаляти та модифікувати дані в реляційних БД [7].

На початку 1970-х років в одній із дослідницьких лабораторій компанії IBM була розроблена експериментальна реляційна СУБД IBM System R, для якої потім була створена спеціальна мова SEQUEL, що дозволяла відносно просто керувати даними в цій СУБД. Метою розробки було створення простої непроцедурної мови, якою міг скористатися будь-який користувач, що навіть не має навичок програмування. Оскільки до початку 1980-х років існувало кілька варіантів СУБД різних виробників, причому, кожен з них мав власну реалізацію мови запитів, було прийнято рішення розробити стандарт мови, який гарантуватиме переносимість ПЗ з однієї СУБД на іншу. Мова SQL була офіційно прийнята групами стандартів ANSI та ISO.9, а набір тестів на відповідність стандарту SQL був розроблений National Institute of Standards and Technology, а також багато продуктів на основі SQL було перевірено цим тестом. Нові стандарти SQL були опубліковані в 2011, 2016 роках. SQL - сукупність операторів, інструкцій та функцій. Оператори SQL розподіляються на:

а) визначення даних (DDL). Оператори які створюють об'єкти (БД, таблиці), видаляють та модифікують: CREATE, DROP, ALTER;

б) маніпуляції даними (DML). Оператори вибору, додавання, змін та видалення даних: SELECT, INSERT, UPDATE, DELETE;

в) управління транзакціями (TCL). Оператори застосування, відкату та розділу транзакцій: COMMIT, ROLLBACK, SAVEPOINT;

г) доступу по даних (DCL). Оператори доступу, відкликання доступу, заборони: GRANT, REVOKE, DENY [8].

2.1.4 Мова програмування PL\SQL

В основі більшої частини Oracle лежить PL/SQL — мова програмування, яка надає процедурні розширення та використовується в Oracle версії SQL, а також служить мовою програмування інструментів коментаря Oracle Developer. PL/SQL виходить на перший план майже у всіх нових продуктах Oracle Corporation. Професійні програмісти використовують PL/SQL для вирішення найрізноманітніших завдань, серед яких:

а) реалізація основної бізнес-логіки в Oracle Server у вигляді процедур, що зберігаються та тригерів БД;

б) генерування документів XML та управління ними виключно на рівні БД;

в) зв'язування вебсторінок із базою даних Oracle;

г) реалізація та автоматизація завдань адміністрування БД - від форми- низькорівневої системи безпеки до управління сегментами відкату у програмах PL/SQL.

Мова PL/SQL створювалася за зразком Ada1 - мови програмування, розробленого для Міністерства оборони. Високорівнева мова Ada приділяє особливу увагу абстракції даних, приховування інформації та інших ключових елементів і стратегій проектування. Завдяки грамотним архітектурним

рішенням PL/SQL став потужною мовою, в якій реалізовано багато сучасних елементів процедурних мов:

а) повний діапазон типів даних, від чисел та рядків до складних структур даних (записів - аналогів рядків у реляційних таблицях), колекцій (аналоги масивів у Oracle) та XML_Type (тип для роботи з документами XML у Oracle та засобами PL/SQL);

б) чітка, легко читана блокова структура, що спрощує розширення і супровід - очікування додатків PL/SQL;

в) умовні, циклічні та послідовні керуючі конструкції, включаючи CASE та три різні види циклів;

г) обробники винятків обробки помилок з урахуванням подій;

д) іменовані, придатні для повторного використання елементи коду: функції, процедури, тригери, об'єктні типи (аналоги класів в об'єктно-орієнтованому програмуванні) та пакети (групи логічно-пов'язаних програм та змінних) [9].

PL/SQL має такі визначальні характеристики:

а) висока структурованість, зручність читання і доступність. мова проста в вивченні, а її ключові слова та структура чітко виражають сенс коду. Програміст з досвідом роботи іншими мовами дуже швидко звикне до нового синтаксису;

б) стандартна переносима мова розробки програм для баз даних Oracle. «Написати один раз і використовувати скрізь» - цей основний принцип PL/SQL був відомий задовго до появи мови Java;

в) вбудована мова. PL/SQL не використовується як самостійна мова програмування. Це вбудована мова, що працює тільки в конкретній керуючій системі, де таким чином, програми PL/SQL можна запускати з БД. Також можливе визначення та виконання програм PL/SQL із форми або звіту Oracle Developer (клієнтський PL/SQL). Однак не має можливості створити

виконуваний файл програми на PL/SQL і запускати його автономно;

г) високопродуктивна, високо інтегрована мова БД. Зараз існує багато способів написання програм, що працюють з БД Oracle. Наприклад, можна використовувати Java та JDBC або Visual Basic і ODBC, а можна скористатися, скажімо, Delphi, C++ і т.д. Код для роботи з БД Oracle простіше написати на PL/SQL, ніж на будь - якій іншій мові програмування. Зокрема, Oracle має кілька розширень, призначених спеціально для PL/SQL, таких як інструкція FORALL, дозволяє підвищити продуктивність обробки запитів на порядок і більше [10].

Мова програмування PL\SQL, якою б потужною вона не була, не має гнучкості і можливостей для створення повноцінних програм. В той же час залишаючись незалежною від ОС оточення Oracle, дозволяє розробляти високоефективні програми, які задовольняють потреби користувачів. З виходом кожної нової версії бази даних Oracle у PL/SQL вводилися все нові фундаментальні удосконалення. Було розроблено безліч зовнішніх і вбудованих пакетів, що розширюють функціональність PL/SQL. Зокрема, мова була доповнена об'єктно-орієнтованими можливостями, різноманітними структурами даних, удосконалений компілятор оптимізував код та видавав попередження про можливі проблеми з якістю та швидкодією коду. В цілому мова була помітно поглиблена і розширена.

2.2 Архітектура

Якщо загалом дивитись на проблему поставлену у дипломній роботі, то можна зрозуміти, що нам необхідно взяти дані з одного місця та зробити з ними деякі умовні дії і перемістити у інше місце. Тобто у деякому сенсі ми займаємось інтеграцією даних, не важливо переміщаємо ми дані за допомогою SQL скрипту або іншого скрипту – це все є інтеграцією даних. Для початку

нам необхідно взяти дані, якщо поглянути на технології вилучення даних, то можемо побачити що вони поділяються на:

а) читання даних з файлів:

- 1) файлів простої структури - з полями фіксованої довжини, з роздільниками;
- 2) файлів складної структури – XML, стандарти форматів файлів – EDIFACT, ACORD, SWIFT тощо;
- 3) текстових файлів без структури чи складної нестандартної структури.

б) читання даних із БД:

- 1) SQL;
- 2) засоби вивантаження\завантаження БД;
- 3) журналів БД за допомогою технології Change Data Capture.

в) вебсервіси;

г) повідомлення з черги даних;

д) об'єкти систем (SQL, OEBS);

е) використання функцій API або зовнішніх процедур для вивантаження\завантаження;

є) читання потокової інформації (стримінг) [11].

У нашому випадку ми використовуємо файли простої структури з полями фіксованої довжини де роздільником є символ пробілу. Також треба виділити те, що інтеграційний процес має ітераційний порядок, наприклад, рисунок 2.3:

а) проблеми відбуваються в процесі інтеграції даних через помилки в якості даних, несумісності та нелогічності даних:

- 1) втрачені значення;
- 2) дублюючі значення;
- 3) надлишкові данні;
- 4) невірна довжина записів;

5) незвичайні величини;

6) невірне розподілення даних.

б) розробники повинні будувати та доопрацьовувати процедури трансформації нескінченну кількість разів для вирішення винятків;

в) в середньому метод спроб і помилок вимагає близько 10 ітерацій, перш ніж досягти успіху в інтеграції [12].

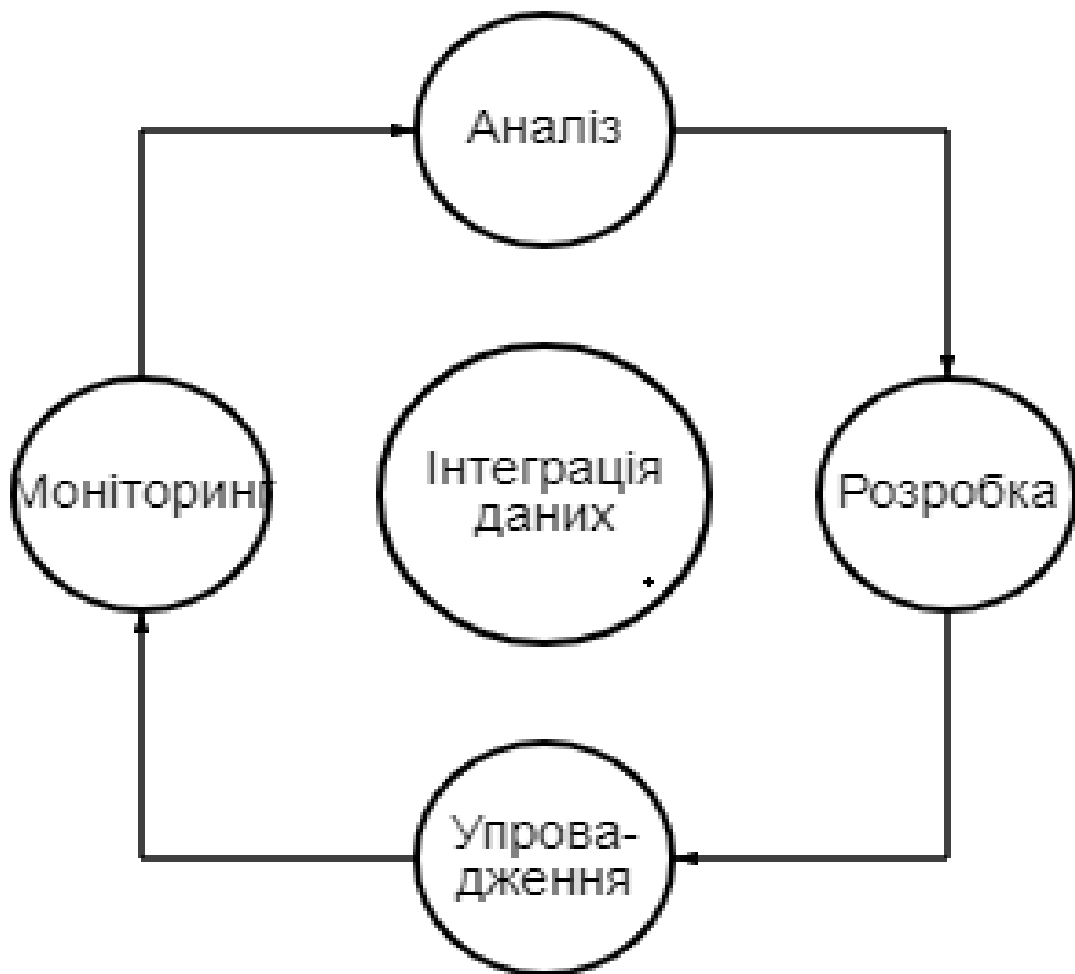


Рисунок 2.3 – Схема інтеграції даних [12]

Якщо розглядати процес інтеграції даних за наших умов, то ми маємо файли даних, які потрібно перетворити до виду транзакцій, які зберігаються в

таблицях БД. Побачити схему алгоритму трансформації даних можна на рисунку 2.4 нижче. Тобто вихідні файли даних потрібно перетворити на транзакції в таблиці, робимо це за допомогою утиліти SQL Loader та контроль файлів, які кажучи простою мовою розрізняють символ пробілу у строчці тим самим структурує дані за допомогою чого ми можемо завантажити їх в поля таблиці. Перетворені до необхідного формату вони зберігаються в тимчасових таблицях і за допомогою нашого пакету трансформуються до кінцевого вигляду у якому вони можуть взаємодіяти з іншими системами та таблицями.

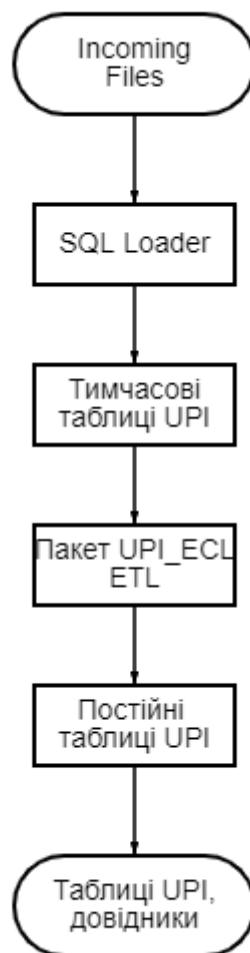


Рисунок 2.4 – Схема алгоритму трансформації даних

Розглядаючи цей самий процес, але вже трохи глибше, то насправді файлів приходить набагато більше ніж вказано у дипломній роботі. Файли з курсами валют, та файли зі звітами про закриття операційного дня, тощо, приклад на рисунку 2.5 нижче. Якщо не поглиблюватись в деталі та пропустити деякі блоки обробки транзакцій між банком еквайером та банком емітентом, між центрами обробки даних та компаніями які являються для них дочірніми, то в цілому обробка в рамках однієї компанії буде виглядати так, як вказано нижче.

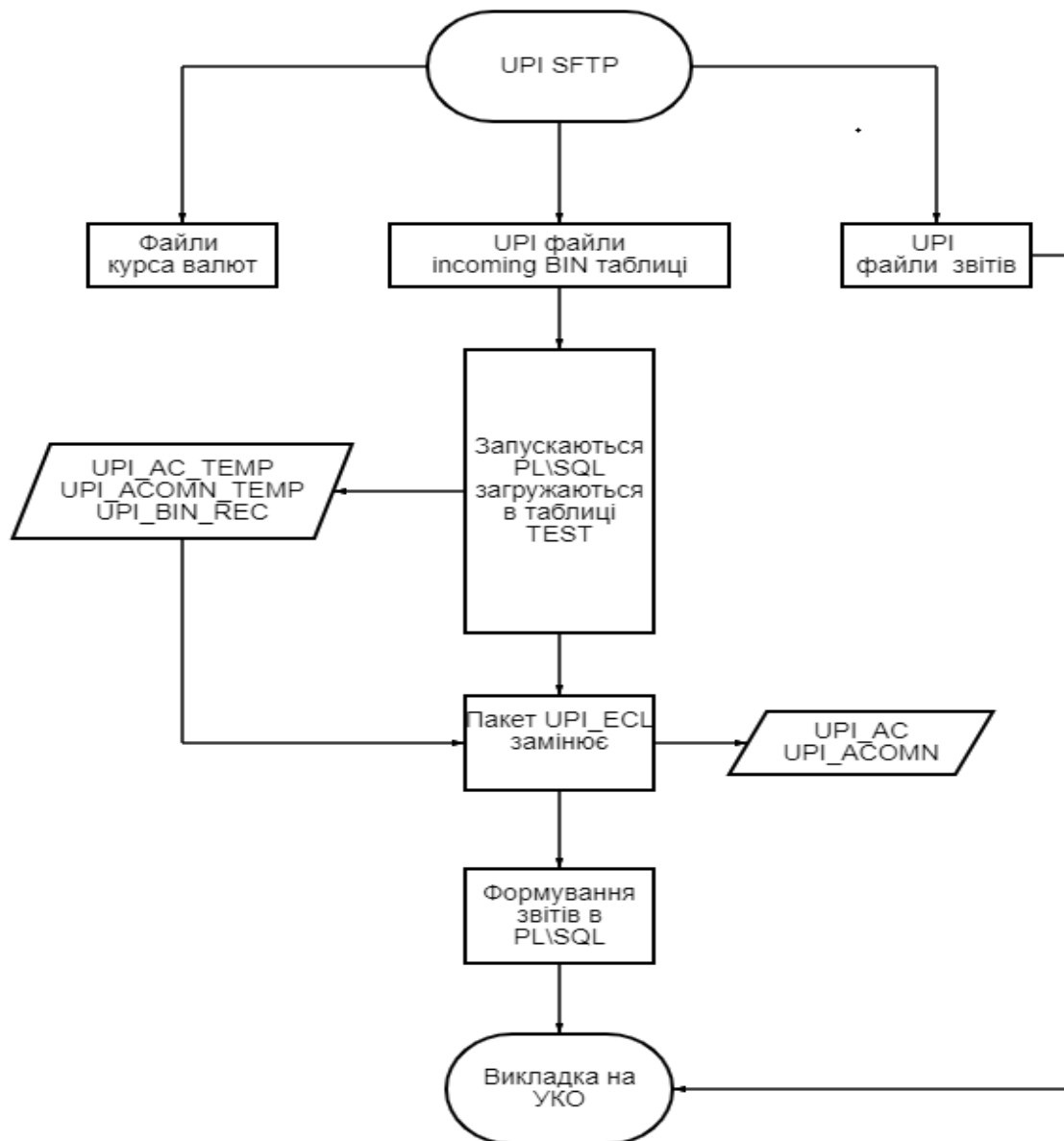


Рисунок 2.5 – Схема функціонування БД

3 РЕАЛІЗАЦІЯ МОДУЛЮ БД

3.1 Опис функціональності системи

На початку створення дипломної роботи було визначено функціонал модулю БД, необхідний для обробки клірингових даних. Необхідні функції модуля:

- а) створення таблиць БД;
- б) створення контроль файлів для завантаження даних;
- в) розробка специфікації пакета;
- г) розробка процедур БД;
- д) розробка функцій БД.

Про реалізацію цих функцій і йтиметься у наступних пунктах.

3.2 Розробка модулю БД

3.2.1 Вхідні файли для обробки з UPI

У БД створимо користувача та схему TEST. Створимо необхідні права користувача для створення, видалення, зміни: таблиць, індексів, пакетів, процедур, функцій та інших об'єктів для роботи програми. З компанії UPI по протоколу FTPS приходять файли даних, поля даних розділені пробілами згідно специфікацій. До файлів даних відносяться:

- а) ACOMN;
- б) AC;
- в) BIN;
- г) IFEEEDTL.

ACOMN файл. Цей формат застосовується до аудиторських трейлерів міжнародних транзакцій. Правила присвоєння імен для аудиту - за типом транзакції. Назва полів і їх значення:

а) з першого по одинадцяте значення: IFDYMMDD??, I – вхідний файл від GSCS, F – файл системи UPI, D - описує деталі транзакцій, YYMMDD?? – 2 символи для ріка, місяця, дня, ?? (01, 02) - номер файлу протягом поточного дня;

б) дванадцяте значення: I – Емітент, A – Еквайер;

в) з тринадцятого по вісімнадцяте значення: COM/COMN/COMB/COMNB – загальна угода по карткам міжнародних банків.

Наприклад файл IFD22033101ACOMN, на рисунку 3.1. I – вхідний файл, F – для системи UPI, D – містить деталі транзакцій, 220331 – 31 березня 2022 року, 01 – перший файл за 31 березня 2022 року, ACOMN – транзакція еквайера по міжнародним картам.

```

..
ifd22033101acomn
ifd22033101aerr
ifd22033101aerrn
ifd22033101afeedt1
ifd22033101fcp
ifd22033101fcpl
ifd22033101icomn
ifd22033101ierr
ifd22033101ierrn
ifd22033101ifeedt1
ifd22033101istifeel
ifl22033101ac
ifl22033101ic
ifo22033101rat
ifo22033106bin
ifr22033101c602dz-980-840
ifr22033101c605dz
ifr22033101c618dz
ifr22033101c702dz-980-840
ifr22033101c702dz-980-980
ifr22033101c708dz
ifr22033101c712dz
ifr22033101c713dz

```

Рисунок 3.1 – Приклад вхідного файлу IFD22033101ACOMN

АС файл. Цей формат застосовується до аудиторських трейлерів українських транзакцій. Правила присвоєння імен для аудиту - за типом транзакції. Назва полів і їх значення:

а) з першого по одинадцяте значення: IFLYYMMDD??, I – вхідний файл від GSCS, F – файл системи UPI, L – локальна транзакція, YYMMDD?? – 2 символу для ріка, місяця, дня, ?? (01, 02) - номер файлу протягом поточного дня;

б) дванадцяте значення: I – Емітент, A – Еквайер;

в) тринадцяте: C – загальна угода по карткам українських банків.

Наприклад файл IFL22041301AC, на рисунку 3.2. I – вхідний файл, F – для системи UPI, L – локальна транзакція, 220413 – 13 квітня 2022 року, 01 – перший файл за 13 квітня 2022 року, AC – транзакція еквайера по українським карткам.

```
..
ifd22041301acom
ifd22041301acomn
ifd22041301aerr
ifd22041301aerrn
ifd22041301afeedt1
ifd22041301fcp
ifd22041301fcpl
ifd22041301icom
ifd22041301icomn
ifd22041301ierr
ifd22041301ierrn
ifd22041301ifeedt1
ifl22041301ac
ifl22041301ic
ifo22041301rat
ifo22041306bin
ifr22041301c602dz-980-840
ifr22041301c605dz
ifr22041301c612dz
ifr22041301c618dz
ifr22041301c702dz-980-840
ifr22041301c702dz-980-980
ifr22041301c708dz
ifr22041301c712dz
ifr22041301c713dz
```

Рисунок 3.2 – Приклад вхідного файлу IFL22041301AC

VIN файли. Файли оновлень, які містять список фінансових організацій, які є членами UPI. Вміст такого файлу приводиться на рисунку 3.3.

H08202204190A1900					
03020000	Citic Bank forCredit Card	10156	00356390	16C	
03020000	Citic Bank forCredit Card	30156	00356391	16C	
03020000	Citic Bank forCredit Card	40156	00356392	16C	
04012902	Bank of Shanghai	00156	00356827	16C	
04012902	Bank of Shanghai	00156	00356828	16C	
04012902	Bank of Shanghai	00156	00356829	16C	
04012902	Bank of Shanghai	00156	00356830	16C	
01040000	BANK OF CHINA	00156	00356833	16C	
01040000	BANK OF CHINA	00156	00356835	16C	
01030000	Credit Card Center of China Everbright Bank	00156	00356839	16C	
01030000	Credit Card Center of China Everbright Bank	00156	00356840	16C	
01100000	Shanghai Pudong Development Bank	00156	00356850	16C	
01100000	Shanghai Pudong Development Bank	00156	00356851	16C	
03050001	China Minsheng Bank Credit Card (CUP Data)	00156	00356855	16C	
03050001	China Minsheng Bank Credit Card (CUP Data)	00156	00356857	16C	
03050001	China Minsheng Bank Credit Card (CUP Data)	00156	00356858	16C	
03050001	China Minsheng Bank Credit Card (CUP Data)	00156	00356859	16C	
06105840	Ping An Bank Company Ltd.	30156	00356868	16C	
06105840	Ping An Bank Company Ltd.	00156	00356869	16C	
01020000	Industrial and Commercial bank of China	10156	00356879	16C	
01020000	Industrial and Commercial bank of China	10156	00356880	16C	
01020000	Industrial and Commercial bank of China	30156	00356881	16C	
01020000	Industrial and Commercial bank of China	30156	00356882	16C	
03080000	China Merchants bank	00156	00356889	16C	
03080000	China Merchants bank	00156	00356890	16C	
01050999	China Construction bank	00156	00356905	16C	
01020000	Industrial and Commercial bank of China	10156	00360883	14C	
01020000	Industrial and Commercial bank of China	30156	00360884	14C	
01020000	Industrial and Commercial bank of China	00156	00370246	15C	
01020000	Industrial and Commercial bank of China	00156	00370247	15C	
01020000	Industrial and Commercial bank of China	00156	00370248	15C	
01020000	Industrial and Commercial bank of China	00156	00370249	15C	
01020000	Industrial and Commercial bank of China	00156	00370267	16C	
03080010	China Merchants bank	00156	00370285	15C	
03080010	China Merchants bank	00156	00370286	15C	
03080010	China Merchants bank	00156	00370287	15C	
03080010	China Merchants bank	00156	00370289	15C	
01020000	Industrial and Commercial bank of China	00156	00374738	15C	
01020000	Industrial and Commercial bank of China	40156	00374739	15C	
03020000	Citic Bank forCredit Card	00156	00376966	15C	
03020000	Citic Bank forCredit Card	00156	00376968	15C	
03050001	China Minsheng Bank Credit Card (CUP Data)	10156	00377152	15C	
03050001	China Minsheng Bank Credit Card (CUP Data)	30156	00377153	15C	
03050001	China Minsheng Bank Credit Card (CUP Data)	40156	00377155	15C	
03050001	China Minsheng Bank Credit Card (CUP Data)	50156	00377158	15C	
01100000	Shanghai Pudong Development Bank	00156	00377617	15C	
01040000	BANK OF CHINA	00156	00377677	15C	
03020000	Citic Bank forCredit Card	00156	00400360	16C	
04012902	Bank of Shanghai	00156	00402673	16C	
04012902	Bank of Shanghai	00156	00402674	16C	
01020000	Industrial and Commercial bank of China	00156	00402701	16C	
01030000	Agriculture bank of China	00156	00403361	16C	
03020000	Citic Bank forCredit Card	00156	00403391	16C	
03020000	Citic Bank forCredit Card	00156	00403392	16C	
03020000	Citic Bank forCredit Card	00156	00404393	16C	
01030000	Agriculture bank of China	00156	00404117	16C	
01030000	Agriculture bank of China	00156	00404119	16C	
01030000	Agriculture bank of China	00156	00404120	16C	
03020000	Citic Bank forCredit Card	00156	00404917	16C	

Рисунок 3.3 Зміст файлу BIN

AFEEDTL файл - файл розрахунку комісії UPI, детальний аудиторський трейлер для файлу розрахунку комісій для учасників, включаючи докладну інформацію про комісійну винагороду за обмін (IRF) та сервісний збір. Приклад файлу під ім'ям IFD2204130IFEEDTL на рисунку 3.2

3.2.2 Розробка таблиць БД

Для завантаження даних із вхідних файлів створюємо таблиці:

- 6) UPI AC TEMP;

в) UPI_AFEEDTL;

г) UPI_BIN_HEADER.

З тимчасових таблиць - UPI_ACOMN_TEMP, UPI_AC_TEMP дані зберігаються в таблиці UPI_ACOMN, UPI_AC у яких дані перетворюються на лету, використовуючи технологію ETL, в додатку А приведен код скриптів на створення таблиць згідно специфікацій.

3.2.3 Створення контроль файлів для завантаження даних

Для завантаження даних із файлів даних у БД Oracle є утиліта - SQL Loader. Вхідні файли з UPI перейменовуються з розширенням (*.dat). У додатку Б наведено вихідний код control файлів які для завантаження даних використовують раніше перейменовані (*.dat) файли, загальний принцип роботи з SQL Loader наведено на рисунку 2.1. Наприклад, для завантаження даних у таблицю UPI_ACOMN_TEMP з командного рядка запускаємо команду: `sqlldr test/test@nsrkc control='UPI_ACOMN_TEMP.ctl'`

Коли дані були завантажені в таблицю перевіряємо наявність даних у таблиці, через Toad For Oracle, приклад на рисунку 3.4. Після завантаження даних у таблицю, створюється UPI_ACOMN_TEMP.log, де ми можемо перевірити вдале або не вдале завантаження даних. У разі не вдалого завантаження даних у таблицю створюється файл з ім'ям файлу (*.dat) і з розширенням (*.bad), куди поміщаються не завантажені транзакції.

Table: UPI_ACOMN_TEMP

ACQUIRING_IC	FORWARDING_IC	SYSTEM_TRACE_AUDIT_NUMBER	TRANSMISSION_DATE	PAN	AMOUNT_TRANSACTION	MESSAGE_TYPE	PROCESSING_CODE	MERCHANT_TYPE	CARD_ACCEPTOR_TERMINAL_ID	CARD_ACCEPTOR_IC
44430804	00049992	643152	0325235901	6250947000000014	000000009000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	643652	0326002052	6250947000000014	000000006500	0200	200000	5311	01080209	3810000500000001
44430804	00049992	643277	0326001154	6250947000000014	000000007200	0200	200000	5311	01080209	3810000500000001
44430804	00049992	587400	0326002919	6250947000000014	000000008500	0200	200000	5311	01080209	3810000500000001
44430804	00049992	586900	0326000753	6250947000000014	000000007200	0200	000000	5311	01080209	3810000500000001
44430804	00049992	643402	0326001849	6250947000000014	000000007000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	643902	0326002646	6250947000000014	000000009900	0200	000000	5311	01080209	3810000500000001
44430804	00049992	353150	0331165657	6250947000000014	000000005000	0200	000000	5311	01080209	3810000500000001

Рисунок 3.4 – Приклад завантаження даних у таблицю

3.2.4 Розробка специфікації пакету БД

Пакет складається з двох частин – специфікації та тіла. Специфікація є обов'язковою частиною і визначає, як розробник може використовувати пакет: які програми можна викликати, які курсори відкривати тощо. Тіло пакета — необов'язкова, але майже завжди присутня частина; вона містить код перерахованих у специфікації програм (і, можливо, курсорів), а також інші необхідні елементи коду. Специфікація пакета містить визначення всіх загальнодоступних елементів пакета, які можуть посилатися ззовні. Специфікація нагадує великий розділ оголошень, що не містить блоків PL/SQL або коду, що виконується. З добре спроектованої специфікації можна отримати всю необхідну для використання пакета інформацію. В тілі пакета знаходиться весь код, який необхідний для реалізації елементів, визначених у специфікації пакета. Тіло може містити відсутні у специфікації особисті елементи, на які не можна посилатися ззовні пакета, зокрема оголошення змінних та визначення пакетних модулів. Крім того, у тілі пакета може бути

виконуваний ініціалізаційний розділ, який виконується лише один раз для ініціалізації пакета. Концепція ініціалізації добре відома, однак у контексті пакетів вона має особливе значення. В даному випадку ініціалізується не окрема змінна, а весь пакет шляхом виконання коду довільної складності. При цьому Oracle стежить за тим, щоб пакет ініціалізувався лише один раз за сеанс [13].

Створюємо пакет із ім'ям TEST.UPI_ECL на схемі TEST (user). TEST – користувач БД Oracle для тестування нашого модуля. У специфікацію пакету TEST.UPI_ECL входять процедури та функції:

- а) Delete_Data_From_Acomn_Temp;
- б) Insert_Data_To_Acomn;
- в) Delete_Data_From_Ac_Temp;
- г) Insert_Data_To_Ac;
- д) Set_Transmission_Date_Time2;
- е) Set_RDATE_ID; Reconsilation;
- є) GET_CNTRY_BY_PAN.

Процедура Delete_Data_From_Acomn_Temp повністю очищає таблицю UPI_Acomn_Temp, після завантаження даних у таблицю UPI_ACOMN, які мають завантажуватися невеликими порціями щодня за допомогою процедури Insert_Data_To_Acomn.

Процедура Delete_Data_From_AC_Temp повністю очищає таблицю UPI_AC_Temp, після завантаження даних у таблицю UPI_AC, які мають завантажуватися невеликими порціями щодня за допомогою процедури Insert_Data_To_AC.

Функція Set_Transmission_Date_Time2 повертає формат дати "YYYYMMDDHH24MISS", який перетворюється з вхідного формату дати "MMDDHH24MISS".

Функція Set_RDATE_ID повертає ідентифікатор операційного дня.

Процедура RECONCILATION виконує звіряння даних між постигнем та клірингом.

Функція GET_CNTRY_BY_PAN за номером рахунку картки повертає регіон фінансової організації.

3.2.5 Розробка процедур БД

Процедура це модуль, що виконує одну або кілька дій. Оскільки виклик процедури в PL/SQL є окремим оператором, що виконується, блок коду PL/SQL може складатися лише з виклику процедури. Процедури відносяться до ключових компонентів модульного коду, що забезпечують оптимізацію і повторне використання програмної логіки. У цій дипломній роботі використовуються нижче наведені процедури, їх приклад наведено у додатку Г.

Процедура RECONCILIATION для заданого ідентифікатора операційного дня оновлює дані для таблиць UPI_ACOMN та arctrsfm на основі набору даних, що повертаються курсором.

Процедура Delete_Data_From_Acomn_Temp видаляє дані з таблиці UPI_ACOMN_Temp за допомогою швидкої операції truncate.

Процедура Delete_Data_From_AC_Temp видаляє дані з таблиці UPI_AC_Temp за допомогою швидкої операції truncate.

Процедура Insert_Data_To_Acomn вставляє дані в таблицю UPI_ACOMN із таблиці UPI_ACOMN_Temp. Також ця процедура використовує механізм ELT для перетворення даних на лету.

Процедура Insert_Data_To_AC вставляє дані в таблицю UPI_AC із таблиці UPI_AC_Temp. Також ця процедура використовує механізм ELT для перетворення даних на лету.

3.2.6 Розробка функцій БД

Функція є модуль, який повертає значення командою RETURN. На відміну від виклику процедури, який уявляє собою окремий оператор, виклик

функції завжди є частиною виконаного оператора, тобто включається у вираз або служить як значення по замовчуванням, що присвоюється змінною при оголошенні. Значення, що повертається функцією, належить до певного типу даних. Функція може використовуватися замість виразу, який має той самий тип даних, що і значення яке вона повертає. Нижче наведені функції які застосовуються в нашому модулі, їх приклад знаходиться в додатку Г.

Функція `Set_Transmission_Date_Time2` повертає формат дати "YYYYMMDDHH24MISS", який перетворюється з вхідного формату дати "MMDDHH24MISS". Функція повертає дату поточного або попереднього року.

Функція `Set_RDATE_ID` повертає ідентифікатор операційного дня. З урахуванням того, що дані приходять з китайським часовим поясом, функція повертає дату операційного дня з українським часовим поясом, який відрізняється на шість годин.

Функція `GET_CNTRY_BY_PAN` за номером рахунку картки повертає регіон фінансової організації.

4 ТЕСТУВАННЯ МОДУЛЮ БД

На даному етапі необхідно переконатись, що розроблений модуль працює коректно, дані з таблиць копіюються у правильному форматі без помилок і всі вищезгадані функції відпрацювали правильно. Спочатку створимо нову сесію з БД, яку можна побачити на рис. 4.1.

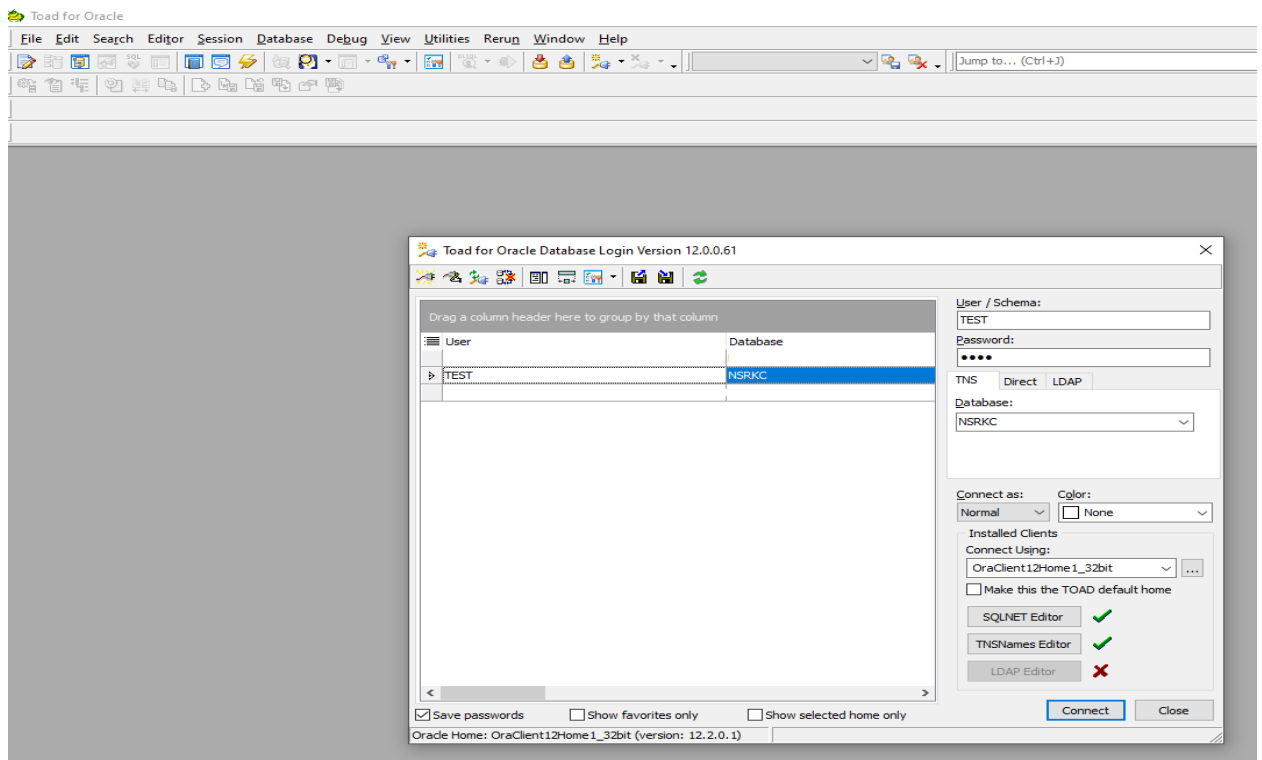


Рисунок 4.1 – Вхід до БД Oracle

Створюємо нову сесію до БД під назвою NSRC під користувачем (schema) TEST, вводимо пароль за умовчанням TEST. Після підключення до БД знайдемо необхідні таблиці які пов'язані з системою UPI. Серед списку об'єктів знаходимо об'єкт таблиці, далі за шаблоном (UPI*) знаходимо всі таблиці, що цікавлять нас. Таблиці БД були створені раніше за допомогою скриптів, приклад наведено на рисунку 4.2. Попередні дані з таблиць були видалені за допомогою процедур які наводяться у додатку А. На рисунку 4.3. можна побачити список таблиць.

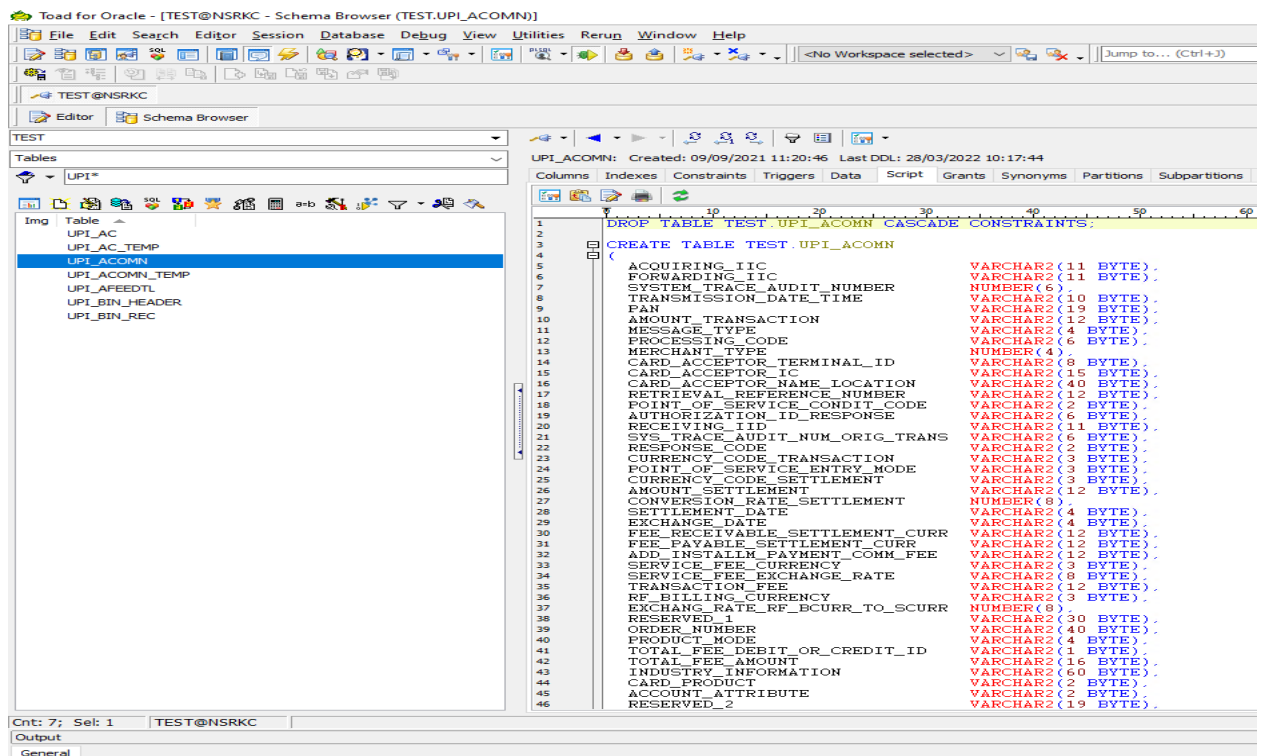


Рисунок 4.2 Створення таблиць БД

The screenshot shows the Toad for Oracle interface with the Schema Browser on the left and the Table List on the right. The Table List contains the following data:

Img	Table	Schema	Tablespace	Last Analyzed	Num Rows	Created	Last DDL
	UPI_AC	TEST	PROD_DATA	28/03/2022 22:00:38	30	09/09/2021 16:36:55	14/04/2022 16:36:24
	UPI_AC_TEMP	TEST	PROD_DATA	05/04/2022 22:04:04		21/02/2022 15:50:03	01/04/2022 13:51:37
	UPI_ACOMN	TEST	PROD_DATA	05/04/2022 22:00:45	210	09/09/2021 11:20:46	28/03/2022 10:17:44
	UPI_ACOMN_TEMP	TEST	PROD_DATA	05/04/2022 22:00:29	8	21/02/2022 15:45:57	01/04/2022 13:51:37
	UPI_AFEEDTL	TEST	PROD_DATA	06/01/2022 22:02:26	11	15/09/2021 17:05:18	23/02/2022 11:20:53
	UPI_BIN_HEADER	TEST	PROD_DATA	10/09/2021 22:00:35	1	10/09/2021 12:16:16	23/02/2022 11:25:34
	UPI_BIN_REC	TEST	PROD_DATA	29/04/2022 22:09:11	4 052	08/09/2021 16:38:15	23/02/2022 11:28:23

Рисунок 4.3 – Список таблиць для системи UPI

Наразі нам необхідно завантажити дані транзакцій за минулий день, які сьогодні прийшли до нас від наших китайських партнерів із міжнародного

клірингового центру. За допомогою контроль файлів описаних вище у пункті 3.2.3 завантажуюмо дані у нашу тимчасову таблицю. На рисунку 4.4 бачимо що дані відображені коректно без зсуву полів таблиці.

The screenshot shows the Toad for Oracle interface with the Schema Browser displaying the TEST database. The table UPI_ACOMN_TEMP is selected, and its data is shown in a grid. The table has 10 columns: ACQUIRING_IC, FORWARDING_IC, SYSTEM_TRACE_AUDIT_NUMBER, TRANSMISSION_DATE, TIME, PAN, AMOUNT_TRANSACTION, MESSAGE_TYPE, PROCESSING_CODE, MERCHANT_TYPE, CARD_ACCEPTOR_TERMINAL_ID, and CARD_ACCEPтор_IC. The data is displayed in a grid with 10 rows and 12 columns. The first row is highlighted in blue.

ACQUIRING_IC	FORWARDING_IC	SYSTEM_TRACE_AUDIT_NUMBER	TRANSMISSION_DATE	TIME	PAN	AMOUNT_TRANSACTION	MESSAGE_TYPE	PROCESSING_CODE	MERCHANT_TYPE	CARD_ACCEPTOR_TERMINAL_ID	CARD_ACCEPтор_IC
44430804	00049992	643152	0325223901		62509470000000014	000000009000	0200	000000	5311	01080209	381000050000001
44430804	00049992	643652	0326002052		62509470000000014	000000006500	0220	200000	5311	01080209	381000050000001
44430804	00049992	643277	0326001154		62509470000000014	000000007200	0220	200000	5311	01080209	381000050000001
44430804	00049992	587400	0326002919		62509470000000014	000000008500	0220	200000	5311	01080209	381000050000001
44430804	00049992	586900	0326000753		62509470000000014	000000007200	0200	000000	5311	01080209	381000050000001
44430804	00049992	643402	0326001849		62509470000000014	000000007000	0200	000000	5311	01080209	381000050000001
44430804	00049992	643902	0326002546		62509470000000014	000000009900	0200	000000	5311	01080209	381000050000001
44430804	00049992	353150	0331165657		62509470000000014	000000005000	0200	000000	5311	01080209	381000050000001

Рисунок 4.4 – Тимчасова таблиця UPI_ACOMN_TEMP

Тепер дані необхідно завантажити у постійну таблицю. За допомогою пакету який був розроблений і згаданий вище у пункті 3.2.4, завантажимо наші дані. Також транзакції будуть відформатовані під потреби компанії за допомогою функцій, що перебувають у цьому пакеті. На рисунку 4.5 ми можемо побачити структуру пакету та саме тіло пакету UPI_ECL. Далі переходимо до нашої постійної таблиці UPI_ACOMN, можемо побачити на рисунку 4.6, що транзакції додалися правильно, а також були відформатовані.

Виходячи з отриманих результатів тесту, вважаю, що модуль БД виконує призначену йому мету на сто відсотків.

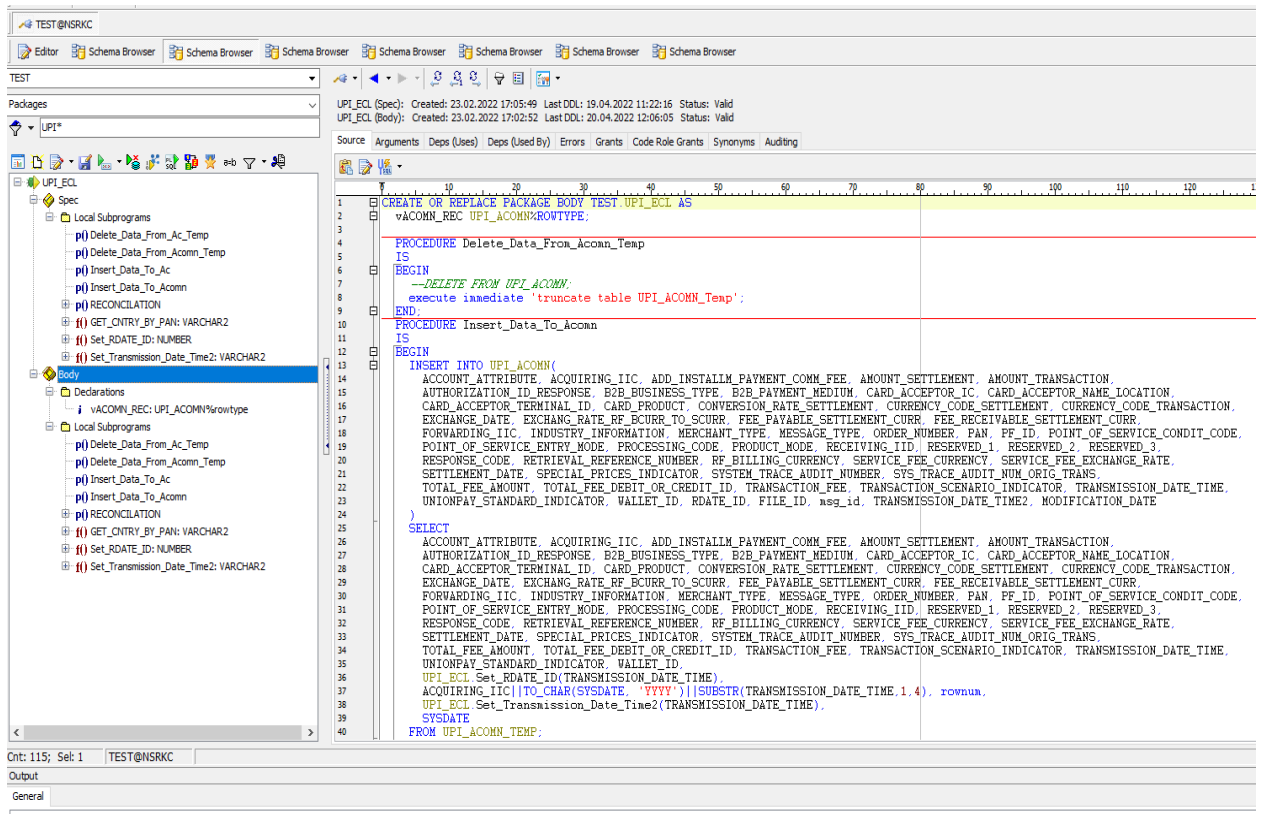


Рисунок 4.5 – Структура пакета UPI_ECL

Toad for Oracle - [TEST@NSRKC - Schema Browser (TEST.UPI_ACOMN)]

File Edit Search Editor Session Database Debug View Utilities Run Window Help

Schema Browser Schema Browser Schema Browser Schema Browser Schema Browser Schema Browser

TEST

Table: UPI*

UPI_ACOMN: Created: 09.09.2021 11:20:46 Last DDL: 28.03.2022 10:17:44

Columns: Indices Constraints Triggers Data Script Grants Synonyms Partitions Subpartitions Status Size Referential Used By Policies Auditing

ACQUIRING_IC	FORWARDING_IC	SYSTEM_TRACE_AUDIT_NUMBER	TRANSMISSION_DATE_TIME	PAN	AMOUNT_TRANSACTION	MESSAGE_TYPE	PROCESSING_CODE	MERCHANT_TYPE	CARD_ACCEPTOR_TERMINAL_ID	CARD_ACCEPTOR_IC
44430804	00049992	357311	0901121815	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	385188	0901131626	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	882186	0902132519	6250947000000014	000000100000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	114276	1129190226	6250947000000014	000000001000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	487151	1301061656	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	953651	1123210516	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	976649	1123159515	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	517149	1124215958	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	535276	1127061400	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	832901	1201172752	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	794526	1201155626	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	152649	1208162732	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	586277	1208162702	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	972401	1213172237	6250947000000014	0000000001500	0200	000000	5311	01080209	3810000500000001
44430804	00049992	474777	1213172754	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	431275	1213172555	6250947000000014	000000000100	0220	200000	5311	01080209	3810000500000001
44430804	00049992	513652	1213203108	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	473652	1213172511	6250947000000014	000000000100	0200	000000	5311	01080209	3810000500000001
44430804	00049992	590399	0208202811	6250947000000014	0000000002300	0200	000000	5311	01080209	3810000500000001
44430804	00049992	534401	0208213829	6250940500000006	000000000500	0200	000000	5311	01080209	3810000500000001
44430804	00049992	501435	0825121143	6250940500000006	0000000001000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	482687	0825115859	6250947000000014	0000000001000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	560935	0825160518	6250947000000014	0000000001000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	735186	0825120436	6250947000000014	0000000009000	0200	000000	5311	01080209	3810000500000001
44430804	00049992	504117	0805174610	6250947000000014	0000000010000	0200	000000	5311	01080209	3810000500000001

Row: 1 of 20 total rows

Ont: 7; Sel: 1 TEST@NSRKC

Output

General

Рисунок 4.6 – Таблица UPI_ACOMN

ВИСНОВКИ

В рамках кваліфікаційної дипломної роботи було розроблено модуль бази даних, який дозволяє автоматично обробляти транзакції міжнародної системи UPI.

Для досягнення мети було використано ряд сервісів:

а) програмне середовище Toad for Oracle - програмне забезпечення, що використовується для роботи з реляційними системами управління базами даних. Крім Oracle Database, підтримується Microsoft SQL Server та іншими системами та хмарними платформами;

б) утиліта SQL Loader - утиліта яка завантажує дані із зовнішніх файлів у таблиці бази даних Oracle. Вона має потужний механізм парсингу даних, який накладає невелике обмеження на формат даних у зовнішньому файлі;

в) мова програмування PL\SQL - мова програмування, процедурне розширення мови SQL, розроблене корпорацією Oracle. Базується мовою Ада.

Створений модуль БД пройшов тестування, на сьогоднішній момент недоліків не виявлено. Він має усі необхідні функції: створення таблиць, видалення та оновлення таблиць.

Практична цінність роботи полягає у тому, що створений модуль БД перетворює транзакції до вигляду міжнародних стандартів, для подальшої обробки даних в процесингових центрах і банках, що в свою чергу заощаджує час та кошти компанії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1 Kyte, T. Arhitecture Overview / T. Kyte, D. Kuhn // Expert Oracle Database Architecture. – New York: Apress Media, 2014. – Section. 2. – P. 53–73.
- 2 Kyte, T. Arhitecture Overview / T. Kyte, D. Kuhn // Expert Oracle Database Architecture. – New York: Apress Media, 2014. – Section. 2. – P. 53–73.
- 3 Kyte, T. Memory Structures / T. Kyte, D. Kuhn // Expert Oracle Database Architecture. – New York: Apress Media, 2014. – Section. 4. – P. 127–173.
- 4 Gennick, J. Introduction to SQL*Loader / J. Gennick, S. Mishra // Oracle SQL*Loader The Definitive Guide. – California: O'Reily Media, 2001. – Section. 1. –P. 1-14.
- 5 Toad for Oracle 12.0. Guide to Using Toad. [Electronic resource]. / Access mode: <https://support.quest.com/toad-for-oracle/16.0/technical-documents>
- 6 Toad for Oracle 12.0. Guide to Using Toad. [Electronic resource]. / Access mode: <https://support.quest.com/toad-for-oracle/16.0/technical-documents>
- 7 Gruber, M. Introduction to SQL / M. Gruber // Understanding SQL. – San Francisco: Sybex, 1990. – Section. 2. – P. 9-17.
- 8 Weinborg, P. An Overview of SQL / P. Weinborg, J. Groff, A. Oppel // SQL The Complete Reference. – New York: McGraw-Hill, 2010. –Section. 1. –P. 17-45.
- 9 Feuerstein, S. Introduction to PL\SQL / S. Feuerstein, B. Pribyl // Oracle PL\SQL Programming. – Sebastopol: O'Reilly Media, 2014. – Section. 1. – P. 43-61.
- 10 Moor, S. Overview of PL\SQL / S. Moor // Oracle Database PL\SQL Language Reference. – California: Oracle Corporation, 2014. – Section. 1. – P. 35-47.
- 11 Jiawei, H. Getting to Know Your Data / H. Jiawei, K. Micheline, P. Jian // Data Mining Concepts and Techniques. – Walthman: Elsevier Inc, 2012. – Section. 2. – P. 40-79.

12 Feuerstein, S. PL\SQL Application Construction / S. Feuerstein, B. Pribyl // Oracle PL\SQL Programming. – Sebastopol: O'Reilly Media, 2014. – Section. 5. – P. 651-685.

13 Soares, S. Organizing for Data Governance / S. Soares // The Chief Data Officer Handbook for Data Governance. – Canada: MC Press Online, 2014. – Section. 3. – P. 48-51.

ДОДАТОК А
Код створення таблиць

DROP TABLE TEST.UPI_ACOMN_TEMP CASCADE CONSTRAINTS;

CREATE TABLE TEST.UPI_ACOMN_TEMP

```
(
ACQUIRING_IIC                                VARCHAR2(11 BYTE),
FORWARDING_IIC                              VARCHAR2(11 BYTE),
SYSTEM_TRACE_AUDIT_NUMBER                    NUMBER(6),
TRANSMISSION_DATE_TIME                      VARCHAR2(10 BYTE),
PAN                                           VARCHAR2(19 BYTE),
AMOUNT_TRANSACTION                           VARCHAR2(12 BYTE),
MESSAGE_TYPE                                VARCHAR2(4 BYTE),
PROCESSING_CODE                              VARCHAR2(6 BYTE),
MERCHANT_TYPE                                NUMBER(4),
CARD_ACCEPTOR_TERMINAL_ID                     VARCHAR2(8 BYTE),
CARD_ACCEPTOR_IC                             VARCHAR2(15 BYTE),
CARD_ACCEPTOR_NAME_LOCATION                  VARCHAR2(40 BYTE),
RETRIEVAL_REFERENCE_NUMBER                   VARCHAR2(12 BYTE),
POINT_OF_SERVICE_CONDIT_CODE                 VARCHAR2(2 BYTE),
AUTHORIZATION_ID_RESPONSE                    VARCHAR2(6 BYTE),
RECEIVING_IID                                VARCHAR2(11 BYTE),
SYS_TRACE_AUDIT_NUM_ORIG_TRANS                VARCHAR2(6 BYTE),
RESPONSE_CODE                                VARCHAR2(2 BYTE),
CURRENCY_CODE_TRANSACTION                     VARCHAR2(3 BYTE),
POINT_OF_SERVICE_ENTRY_MODE                   VARCHAR2(3 BYTE),
CURRENCY_CODE_SETTLEMENT                     VARCHAR2(3 BYTE),
AMOUNT_SETTLEMENT                            VARCHAR2(12 BYTE),
CONVERSION_RATE_SETTLEMENT                     NUMBER(8),
SETTLEMENT_DATE                              VARCHAR2(4 BYTE),
EXCHANGE_DATE                                VARCHAR2(4 BYTE),
FEE_RECEIVABLE_SETTLEMENT_CURR                VARCHAR2(12 BYTE),
FEE_PAYABLE_SETTLEMENT_CURR                   VARCHAR2(12 BYTE),
ADD_INSTALLMENT_PAYMENT_COMM_FEE              VARCHAR2(12 BYTE),
SERVICE_FEE_CURRENCY                         VARCHAR2(3 BYTE),
SERVICE_FEE_EXCHANGE_RATE                     VARCHAR2(8 BYTE),
TRANSACTION_FEE                              VARCHAR2(12 BYTE),
RF_BILLING_CURRENCY                           VARCHAR2(3 BYTE),
EXCHANG_RATE_RF_BCURREN_TO_SCURREN            NUMBER(8),
RESERVED_1                                   VARCHAR2(30 BYTE),
ORDER_NUMBER                                 VARCHAR2(40 BYTE),
PRODUCT_MODE                                 VARCHAR2(4 BYTE),
TOTAL_FEE_DEBIT_OR_CREDIT_ID                  VARCHAR2(1 BYTE),
TOTAL_FEE_AMOUNT                             VARCHAR2(16 BYTE),
INDUSTRY_INFORMATION                         VARCHAR2(60 BYTE),
CARD_PRODUCT                                 VARCHAR2(2 BYTE),
ACCOUNT_ATTRIBUTE                             VARCHAR2(2 BYTE),
```

RESERVED_2	VARCHAR2(19 BYTE),
UNIONPAY_STANDARD_INDICATOR	VARCHAR2(1 BYTE),
B2B_BUSINESS_TYPE	VARCHAR2(2 BYTE),
B2B_PAYMENT_MEDIUM	VARCHAR2(1 BYTE),
WALLET_ID	VARCHAR2(8 BYTE),
PF_ID	VARCHAR2(8 BYTE),
SPECIAL_PRICES_INDICATOR	VARCHAR2(2 BYTE),
TRANSACTION_SCENARIO_INDICATOR	VARCHAR2(3 BYTE),
RESERVED_3	VARCHAR2(115 BYTE)
)	

DROP TABLE TEST.UPI_AC_TEMP CASCADE CONSTRAINTS;

CREATE TABLE TEST.UPI_AC_TEMP	
(	
HEADER_AND_TRANSACTION_CODE	VARCHAR2(13 BYTE),
TRANSMISSION_DATA_TIME	VARCHAR2(10 BYTE),
SYSTEM_TRACE_AUDIT_NUMBER	NUMBER(6),
ACQUIRING_IIC	VARCHAR2(11 BYTE),
FORWARDING_IIC	VARCHAR2(11 BYTE),
MESSAGE_TYPE	VARCHAR2(4 BYTE),
PAN	VARCHAR2(19 BYTE),
TOKEN_NO	VARCHAR2(19 BYTE),
PROCESSING_CODE	VARCHAR2(6 BYTE),
TRANSACTION_AMOUNT	VARCHAR2(12 BYTE),
TRANSACTION_CURRENCY_CODE	VARCHAR2(3 BYTE),
SETTLEMENT_AMOUNT_DC_INDICATOR	VARCHAR2(1 BYTE),
SETTLEMENT_AMOUNT	VARCHAR2(12 BYTE),
SETTLEMENT_CURRENCY_CODE	VARCHAR2(3 BYTE),
SETTLEMENT_EXCHANGE_RATE	NUMBER(8),
AMOUNT_CARDHOLDER_BILLING	VARCHAR2(12 BYTE),
CURR_CODE_CARDHOLDER_BILLING	VARCHAR2(3 BYTE),
CONVER_RATE_CARDHOLDER_BILLING	NUMBER(8),
SETTLEMENT_DATE	VARCHAR2(4 BYTE),
DATE_OF_AUTHORIZATION	VARCHAR2(4 BYTE),
EXCHANGE_DATE	VARCHAR2(4 BYTE),
MERCHANT_TYPE	NUMBER(4),
MERCHANT_COUNTRY_CODE	NUMBER(3),
POINT_OF_SERVICE_ENTRY_MODE_CODE	VARCHAR2(3 BYTE),
CARD_SEQUENCE_NUMBER	VARCHAR2(3 BYTE),
POINT_OF_SERVICE_CONDIT_CODE	VARCHAR2(2 BYTE),
RETRIEVAL_REFERENCE_NUMBER	VARCHAR2(12 BYTE),
AUTHORIZATION_ID_RESPONSE	VARCHAR2(6 BYTE),
RESPONSE_CODE	VARCHAR2(2 BYTE),

CARD_ACCEPTOR_TERMINAL_ID	VARCHAR2(8 BYTE),
CARD_ACCEPTOR_IC	VARCHAR2(15 BYTE),
CARD_ACCEPTOR_NAME_LOCATION	VARCHAR2(40 BYTE),
INSTALLMENT_PAYMENT_TERMS	VARCHAR2(2 BYTE),
MESSAGE_REASON_CODE	VARCHAR2(4 BYTE),
TERMINAL_ENTRY_CAPABILITY	VARCHAR2(1 BYTE),
CHIP_CONDITION_CODE	VARCHAR2(1 BYTE),
TERMINAL_TYPE	VARCHAR2(2 BYTE),
ECI	VARCHAR2(2 BYTE),
TRANSACTION_INITIATION_MODE	VARCHAR2(1 BYTE),
ACCOUNT_ATTRIBUTE	VARCHAR2(2 BYTE),
CARD_LEVEL	VARCHAR2(1 BYTE),
CARD_PRODUCT	VARCHAR2(2 BYTE),
SYSTEM_TRACE_NUMBER_ORIG_TRANS	VARCHAR2(6 BYTE),
SYSTEM_DATE_TIME_ORIG_TRANS	VARCHAR2(10 BYTE),
SETTLEMENT_DATE_ORIG_TRANS	VARCHAR2(4 BYTE),
RECEIVING_IID	VARCHAR2(11 BYTE),
ISSUING_IID	VARCHAR2(11 BYTE),
STAND_IN_AUTHORIZAT_IDENTIFIER	VARCHAR2(1 BYTE),
PAYMENT_SYSTEM_INDICATOR	VARCHAR2(3 BYTE),
ORDER_NUMBER	VARCHAR2(40 BYTE),
CODE_OF_PRICING_SCHEME	VARCHAR2(2 BYTE),
TOTAL_REIM_FEE_AMOUNT	VARCHAR2(16 BYTE),
TOTAL_REIM_FEE_DC_INDICATOR	VARCHAR2(1 BYTE),
TOTAL_REIM_FEE_SETTLEMENT_CURR	VARCHAR2(3 BYTE),
TOTAL_REIM_FEE_BILLING_CURR	VARCHAR2(3 BYTE),
TOTAL_RFER_FROM_RFBC_TO_SCURR	VARCHAR2(8 BYTE),
TOTAL_SERVICE_FEE_AMOUNT	VARCHAR2(16 BYTE),
TOTAL_SERVICE_FEE_DC_INDICATOR	VARCHAR2(1 BYTE),
TOTAL_SERVICE_FEE_SETTLEM_CURR	VARCHAR2(3 BYTE),
TOTAL_SERVICE_FEE_BILLING_CURR	VARCHAR2(3 BYTE),
TOTAL_SFER_FROM_BCURR_TO_SCURR	VARCHAR2(8 BYTE),
PRODUCT_MODE	VARCHAR2(4 BYTE),
INDUSTRY_INFORMATION	VARCHAR2(60 BYTE),
RESERVED	VARCHAR2(300 CHAR)

ДОДАТОК Б
Код керуючих файлів

LOAD DATA CHARACTERSET CL8MSWIN1251

INFILE IFD22041301ACOMN.dat

append

--FILE BODY

INTO TABLE UPI_ACOMN_TEMP

TRAILING NULLCOLS

(
 ACQUIRING_IIC POSITION(1:11),
 FORWARDING_IIC POSITION(13:23),
 SYSTEM_TRACE_AUDIT_NUMBER POSITION(25:30),
 TRANSMISSION_DATE_TIME POSITION(32:41),
 PAN POSITION(43:61),
 AMOUNT_TRANSACTION POSITION(63:74),
 MESSAGE_TYPE POSITION(76:79),
 PROCESSING_CODE POSITION(81:86),
 MERCHANT_TYPE POSITION(88:91),
 CARD_ACCEPTOR_TERMINAL_ID POSITION(93:100),
 CARD_ACCEPTOR_IC POSITION(102:116),
 CARD_ACCEPTOR_NAME_LOCATION POSITION(118:157),
 RETRIEVAL_REFERENCE_NUMBER POSITION(159:170),
 POINT_OF_SERVICE_CONDIT_CODE POSITION(172:173),
 AUTHORIZATION_ID_RESPONSE POSITION(175:180),
 RECEIVING_IID POSITION(182:192),
 SYS_TRACE_AUDIT_NUM_ORIG_TRANS POSITION(194:199),
 RESPONSE_CODE POSITION(201:202),
 CURRENCY_CODE_TRANSACTION POSITION(204:206),
 POINT_OF_SERVICE_ENTRY_MODE POSITION(208:210),
 CURRENCY_CODE_SETTLEMENT POSITION(212:214),
 AMOUNT_SETTLEMENT POSITION(216:227),
 CONVERSION_RATE_SETTLEMENT POSITION(229:236),
 SETTLEMENT_DATE POSITION(238:241),
 EXCHANGE_DATE POSITION(243:246),
 FEE_RECEIVABLE_SETTLEMENT_CURR POSITION(248:259),
 FEE_PAYABLE_SETTLEMENT_CURR POSITION(261:272),
 ADD_INSTALLM_PAYMENT_COMM_FEE POSITION(274:285),
 SERVICE_FEE_CURRENCY POSITION(287:289),
 SERVICE_FEE_EXCHANGE_RATE POSITION(291:298),
 TRANSACTION_FEE POSITION(300:311),
 RF_BILLING_CURRENCY POSITION(313:315),
 EXCHANG_RATE_RF_BCURREN TO_SCURREN POSITION(317:324),
 RESERVED_1 POSITION(326:355),
 ORDER_NUMBER POSITION(357:396),

PRODUCT_MODE POSITION(398:401),
 TOTAL_FEE_DEBIT_OR_CREDIT_ID POSITION(403:403),
 TOTAL_FEE_AMOUNT POSITION(405:420),
 INDUSTRY_INFORMATION POSITION(422:481),
 CARD_PRODUCT POSITION(483:484),
 ACCOUNT_ATTRIBUTE POSITION(486:487),
 RESERVED_2 POSITION(489:507),
 UNIONPAY_STANDARD_INDICATOR POSITION(509:509),
 B2B_BUSINESS_TYPE POSITION(511:512),
 B2B_PAYMENT_MEDIUM POSITION(514:514),
 WALLET_ID POSITION(516:523),
 PF_ID POSITION(525:532),
 SPECIAL_PRICES_INDICATOR POSITION(534:535),
 TRANSACTION_SCENARIO_INDICATOR POSITION(537:539),
 RESERVED_3 POSITION(541:655)
)

LOAD DATA CHARACTERSET CL8MSWIN1251

INFILE IFL22041401AC.dat

--replace

append

--FILE BODY

INTO TABLE UPI_AC_TEMP

(
 HEADER_AND_TRANSACTION_CODE POSITION(1:13),
 TRANSMISSION_DATA_TIME POSITION(15:24),
 SYSTEM_TRACE_AUDIT_NUMBER POSITION(26:31),
 ACQUIRING_IIC POSITION(33:43),
 FORWARDING_IIC POSITION(45:55),
 MESSAGE_TYPE POSITION(57:60),
 PAN POSITION(62:80),
 TOKEN_NO POSITION(82:100),
 PROCESSING_CODE POSITION(102:107),
 TRANSACTION_AMOUNT POSITION(109:120),
 TRANSACTION_CURRENCY_CODE POSITION(122:124),
 SETTLEMENT_AMOUNT_DC_INDICATOR POSITION(126:126),
 SETTLEMENT_AMOUNT POSITION(128:139),
 SETTLEMENT_CURRENCY_CODE POSITION(141:143),
 SETTLEMENT_EXCHANGE_RATE POSITION(145:152),
 AMOUNT_CARDHOLDER_BILLING POSITION(154:165),
 CURR_CODE_CARDHOLDER_BILLING POSITION(167:169),
 CONVER_RATE_CARDHOLDER_BILLING POSITION(171:178),
 SETTLEMENT_DATE POSITION(180:183),
)

DATE_OF_AUTHORIZATION POSITION(185:188),
 EXCHANGE_DATE POSITION(190:193),
 MERCHANT_TYPE POSITION(195:198),
 MERCHANT_COUNTRY_CODE POSITION(200:202),
 POINT_OF_SERVICE_ENTRY_MODE_CODE POSITION(204:206),
 CARD_SEQUENCE_NUMBER POSITION(208:210),
 POINT_OF_SERVICE_CONDITION_CODE POSITION(212:213),
 RETRIEVAL_REFERENCE_NUMBER POSITION(215:226),
 AUTHORIZATION_ID_RESPONSE POSITION(228:233),
 RESPONSE_CODE POSITION(235:236),
 CARD_ACCEPTOR_TERMINAL_ID POSITION(238:245),
 CARD_ACCEPTOR_IC POSITION(247:261),
 CARD_ACCEPTOR_NAME_LOCATION POSITION(263:302),
 INSTALLMENT_PAYMENT_TERMS POSITION(304:305),
 MESSAGE_REASON_CODE POSITION(307:310),
 TERMINAL_ENTRY_CAPABILITY POSITION(312:312),
 CHIP_CONDITION_CODE POSITION(314:314),
 TERMINAL_TYPE POSITION(316:317),
 ECI POSITION(319:320),
 TRANSACTION_INITIATION_MODE POSITION(322:322),
 ACCOUNT_ATTRIBUTE POSITION(324:325),
 CARD_LEVEL POSITION(327:327),
 CARD_PRODUCT POSITION(329:330),
 SYSTEM_TRACE_NUMBER_ORIG_TRANS POSITION(332:337),
 SYSTEM_DATE_TIME_ORIG_TRANS POSITION(339:348),
 SETTLEMENT_DATE_ORIG_TRANS POSITION(350:353),
 RECEIVING_IID POSITION(355:365),
 ISSUING_IID POSITION(367:377),
 STAND_IN_AUTHORIZATION_IDENTIFIER POSITION(379:379),
 PAYMENT_SYSTEM_INDICATOR POSITION(381:383),
 ORDER_NUMBER POSITION(385:424),
 CODE_OF_PRICING_SCHEME POSITION(426:427),
 TOTAL_REIM_FEE_AMOUNT POSITION(429:444),
 TOTAL_REIM_FEE_DC_INDICATOR POSITION(446:446),
 TOTAL_REIM_FEE_SETTLEMENT_CURR POSITION(448:450),
 TOTAL_REIM_FEE_BILLING_CURR POSITION(452:454),
 TOTAL_REFER_FROM_RFBC_TO_SCURR POSITION(456:463),
 TOTAL_SERVICE_FEE_AMOUNT POSITION(465:480),
 TOTAL_SERVICE_FEE_DC_INDICATOR POSITION(482:482),
 TOTAL_SERVICE_FEE_SETTLEMENT_CURR POSITION(484:486),
 TOTAL_SERVICE_FEE_BILLING_CURR POSITION(488:490),
 TOTAL_REFER_FROM_BCURR_TO_SCURR POSITION(492:499),
 PRODUCT_MODE POSITION(501:504),
 INDUSTRY_INFORMATION POSITION(506:565),
 RESERVED POSITION(567:866)

ДОДАТОК В
Код специфікації пакету БД

```
CREATE OR REPLACE PACKAGE TEST.UPI_ECL
PROCEDURE Delete_Data_From_Acomn_Temp;
PROCEDURE Insert_Data_To_Acomn;
PROCEDURE Delete_Data_From_Ac_Temp;
PROCEDURE Insert_Data_To_Ac;
FUNCTION Set_Transmission_Date_Time2 (pTRANSMISSION_DATE_TIME
VARCHAR2) RETURN VARCHAR2;
FUNCTION Set_RDATE_ID (pTRANSMISSION_DATE_TIME varchar2)
RETURN NUMBER;
PROCEDURE RECONCILIATION (pRdate_id number);
FUNCTION GET_CNTRY_BY_PAN(pPan varchar2) RETURN VARCHAR2;
END UPI_ECL;
```

ДОДАТОК Г
Код тіла пакету БД

```
CREATE OR REPLACE PACKAGE BODY TEST.UPI_ECL AS
vACOMN_REC UPI_ACOMN%ROWTYPE;
```

```
PROCEDURE Delete_Data_From_Acomn_Temp
```

```
IS
```

```
BEGIN
```

```
--DELETE FROM UPI_ACOMN;
```

```
execute immediate 'truncate table UPI_ACOMN_Temp';
```

```
END;
```

```
PROCEDURE Insert_Data_To_Acomn
```

```
IS
```

```
BEGIN
```

```
INSERT INTO UPI_ACOMN(ACCOUNT_ATTRIBUTE,
ACQUIRING_IIC,ADD_INSTALLM_PAYMENT_COMM_FEE,
AMOUNT_SETTLEMENT, AMOUNT_TRANSACTION,
AUTHORIZATION_ID_RESPONSE,B2B_BUSINESS_TYPE,
B2B_PAYMENT_MEDIUM,CARD_ACCEPTOR_IC,
CARD_ACCEPTOR_NAME_LOCATION,
CARD_ACCEPTOR_TERMINAL_ID,
CARD_PRODUCT,CONVERSION_RATE_SETTLEMENT,
CURRENCY_CODE_SETTLEMENT, CURRENCY_CODE_TRANSACTION,
EXCHANGE_DATE,EXCHANG_RATE_RF_BCURREN_TO_SCURREN,
FEE_PAYABLE_SETTLEMENT_CURR,
FEE_RECEIVABLE_SETTLEMENT_CURR,
FORWARDING_IIC,INDUSTRY_INFORMATION,MERCHANT_TYPE,
MESSAGE_TYPE,ORDER_NUMBER,PAN,PF_ID,
POINT_OF_SERVICE_CONDIT_CODE,
POINT_OF_SERVICE_ENTRY_MODE,
PROCESSING_CODE, PRODUCT_MODE,
RECEIVING_IID, RESERVED_1, RESERVED_2, RESERVED_3,
RESPONSE_CODE,RETRIEVAL_REFERENCE_NUMBER,
RF_BILLING_CURRENCY,SERVICE_FEE_CURRENCY,
SERVICE_FEE_EXCHANGE_RATE,
SETTLEMENT_DATE,SPECIAL_PRICES_INDICATOR,
SYSTEM_TRACE_AUDIT_NUMBER,
SYS_TRACE_AUDIT_NUM_ORIG_TRANS,
TOTAL_FEE_AMOUNT,TOTAL_FEE_DEBIT_OR_CREDIT_ID,
TRANSACTION_FEE,TRANSACTION_SCENARIO_INDICATOR,
TRANSMISSION_DATE_TIME,
UNIONPAY_STANDARD_INDICATOR, WALLET_ID, RDATE_ID, FILE_ID,
msg_id, TRANSMISSION_DATE_TIME2, MODIFICATION_DATE)
```

```
SELECT
```

```

ACCOUNT_ATTRIBUTE,ACQUIRING_IIC,
ADD_INSTALLMENT_PAYMENT_COMM_FEE,AMOUNT_SETTLEMENT,
AMOUNT_TRANSACTION,
AUTHORIZATION_ID_RESPONSE,B2B_BUSINESS_TYPE,
B2B_PAYMENT_MEDIUM,CARD_ACCEPTOR_IC,
CARD_ACCEPTOR_NAME_LOCATION,
CARD_ACCEPTOR_TERMINAL_ID,CARD_PRODUCT,
CONVERSION_RATE_SETTLEMENT,CURRENCY_CODE_SETTLEMENT,
CURRENCY_CODE_TRANSACTION,
EXCHANGE_DATE,EXCHANGE_RATE_RF_BCURRENTO_SCURREN,
FEE_PAYABLE_SETTLEMENT_CURR,
FEE_RECEIVABLE_SETTLEMENT_CURR,
FORWARDING_IIC,INDUSTRY_INFORMATION,MERCHANT_TYPE,
MESSAGE_TYPE,ORDER_NUMBER,PAN,PF_ID,
POINT_OF_SERVICE_CONDITION_CODE,
POINT_OF_SERVICE_ENTRY_MODE,PROCESSING_CODE,
PRODUCT_MODE,RECEIVING_IID,RESERVED_1,
RESERVED_2,RESERVED_3,
RESPONSE_CODE,RETRIEVAL_REFERENCE_NUMBER,
RF_BILLING_CURRENCY,SERVICE_FEE_CURRENCY,
SERVICE_FEE_EXCHANGE_RATE,
SETTLEMENT_DATE,SPECIAL_PRICES_INDICATOR,
SYSTEM_TRACE_AUDIT_NUMBER,
SYS_TRACE_AUDIT_NUM_ORIG_TRANS,
TOTAL_FEE_AMOUNT,TOTAL_FEE_DEBIT_OR_CREDIT_ID,
TRANSACTION_FEE,TRANSACTION_SCENARIO_INDICATOR,
TRANSMISSION_DATE_TIME,
UNIONPAY_STANDARD_INDICATOR,WALLET_ID,
UPI_ECL.Set_RDATE_ID(TRANSMISSION_DATE_TIME),
ACQUIRING_IIC||TO_CHAR(SYSDATE,
'YYYY')||SUBSTR(TRANSMISSION_DATE_TIME,1,4),rownum,
UPI_ECL.Set_Transmission_Date_Time2(TRANSMISSION_DATE_TIME),
SYSDATE
FROM UPI_ACOMN_TEMP;
END;

```

```

PROCEDURE Delete_Data_From_Ac_Temp
IS
BEGIN
--DELETE FROM UPI_AC;
execute immediate 'truncate table UPI_AC_Temp';
END;

```

```

PROCEDURE Insert_Data_To_Ac
IS

```

```

BEGIN
INSERT INTO UPI_AC(ACCOUNT_ATTRIBUTE,
ACQUIRING_IIC,AMOUNT_CARDHOLDER_BILLING,
AUTHORIZATION_ID_RESPONSE,
CARD_ACCEPTOR_IC,CARD_ACCEPTOR_NAME_LOCATION,
CARD_ACCEPTOR_TERMINAL_ID, CARD_LEVEL,
CARD_PRODUCT, CARD_SEQUENCE_NUMBER,
CHIP_CONDITION_CODE, CODE_OF_PRICING_SCHEME,
CONVER_RATE_CARDHOLDER_BILLING,
CURR_CODE_CARDHOLDER_BILLING,
DATE_OF_AUTHORIZATION, ECI,
EXCHANGE_DATE, FORWARDING_IIC,
HEADER_AND_TRANSACTION_CODE, INDUSTRY_INFORMATION,
INSTALLMENT_PAYMENT_TERMS,ISSUING_IID,
MERCHANT_COUNTRY_CODE, MERCHANT_TYPE,
MESSAGE_REASON_CODE, MESSAGE_TYPE, ORDER_NUMBER,
PAN, PAYMENT_SYSTEM_INDICATOR,
POINT_OF_SERVICE_CONDIT_CODE,
POINT_OF_SERVI_ENTRY_MODE_CODE, PROCESSING_CODE,
PRODUCT_MODE, RECEIVING_IID, RESERVED, RESPONSE_CODE,
RETRIEVAL_REFERENCE_NUMBER,
SETTLEMENT_AMOUNT,SETTLEMENT_AMOUNT_DC_INDICATOR,
SETTLEMENT_CURRENCY_CODE, SETTLEMENT_DATE,
SETTLEMENT_DATE_ORIG_TRANS,SETTLEMENT_EXCHANGE_RATE,
STAND_IN_AUTHORIZAT_IDENTIFIER,
SYSTEM_DATE_TIME_ORIG_TRANS,SYSTEM_TRACE_AUDIT_NUMBER,
SYSTEM_TRACE_NUMBER_ORIG_TRANS,
TERMINAL_ENTRY_CAPABILITY,TERMINAL_TYPE,TOKEN_NO,
TOTAL_REIM_FEE_AMOUNT,
TOTAL_REIM_FEE_BILLING_CURR,
TOTAL_REIM_FEE_DC_INDICATOR,
TOTAL_REIM_FEE_SETTLEMENT_CURR,
TOTAL_RFER_FROM_RFBC_TO_SCURR,
TOTAL_SERVICE_FEE_AMOUNT,
TOTAL_SERVICE_FEE_BILLING_CURR,
TOTAL_SERVICE_FEE_DC_INDICATOR,
TOTAL_SERVICE_FEE_SETTLEM_CURR,
TOTAL_SFER_FROM_BCURR_TO_SCURR,TRANSACTION_AMOUNT,
TRANSACTION_CURRENCY_CODE,
TRANSACTION_INITIATION_MODE, TRANSMISSION_DATA_TIME
)
SELECT
ACCOUNT_ATTRIBUTE, ACQUIRING_IIC,
AMOUNT_CARDHOLDER_BILLING, AUTHORIZATION_ID_RESPONSE,

```

```

CARD_ACCEPTOR_IC,CARD_ACCEPTOR_NAME_LOCATION,
CARD_ACCEPTOR_TERMINAL_ID, CARD_LEVEL,
CARD_PRODUCT, CARD_SEQUENCE_NUMBER,
CHIP_CONDITION_CODE, CODE_OF_PRICING_SCHEME,
CONVER_RATE_CARDHOLDER_BILLING,
CURR_CODE_CARDHOLDER_BILLING, DATE_OF_AUTHORIZATION,
ECI,EXCHANGE_DATE,FORWARDING_IIC,
HEADER_AND_TRANSACTION_CODE, INDUSTRY_INFORMATION,
INSTALLMENT_PAYMENT_TERMS,ISSUING_IID,
MERCHANT_COUNTRY_CODE, MERCHANT_TYPE,
MESSAGE_REASON_CODE, MESSAGE_TYPE,
ORDER_NUMBER, PAN, PAYMENT_SYSTEM_INDICATOR,
POINT_OF_SERVICE_CONDIT_CODE,
POINT_OF_SERVICE_ENTRY_MODE_CODE, PROCESSING_CODE,
PRODUCT_MODE,RECEIVING_IID,RESERVED,RESPONSE_CODE,
RETRIEVAL_REFERENCE_NUMBER,
SETTLEMENT_AMOUNT,      SETTLEMENT_AMOUNT_DC_INDICATOR,
SETTLEMENT_CURRENCY_CODE, SETTLEMENT_DATE,
SETTLEMENT_DATE_ORIG_TRANS, SETTLEMENT_EXCHANGE_RATE,
STAND_IN_AUTHORIZAT_IDENTIFIER,
SYSTEM_DATE_TIME_ORIG_TRANS,SYSTEM_TRACE_AUDIT_NUMBER,
SYSTEM_TRACE_NUMBER_ORIG_TRANS,
TERMINAL_ENTRY_CAPABILITY,TERMINAL_TYPE,TOKEN_NO,
TOTAL_REIM_FEE_AMOUNT,
TOTAL_REIM_FEE_BILLING_CURR,TOTAL_REIM_FEE_DC_INDICATOR,
TOTAL_REIM_FEE_SETTLEMENT_CURR,
TOTAL_RFER_FROM_RFBC_TO_SCURR,
TOTAL_SERVICE_FEE_AMOUNT,
TOTAL_SERVICE_FEE_BILLING_CURR,
TOTAL_SERVICE_FEE_DC_INDICATOR,
TOTAL_SERVICE_FEE_SETTLEM_CURR,
TOTAL_SFER_FROM_BCURR_TO_SCURR,TRANSACTION_AMOUNT,
TRANSACTION_CURRENCY_CODE,
TRANSACTION_INITIATION_MODE, TRANSMISSION_DATA_TIME
FROM UPI_AC_TEMP;
END;

```

```

FUNCTION Set_Transmission_Date_Time2
(pTRANSMISSION_DATE_TIME varchar2) RETURN VARCHAR2
IS
-- 0830172124
-- 20210830172124
v_cur_year varchar2(10) := to_char(sysdate, 'YYYY');
v_date date;
v_format varchar2(20) := 'YYYYMMDDHH24MISS';

```

```

BEGIN
v_date := to_date(v_cur_year|| pTRANSMISSION_DATE_TIME, v_format);
return to_char(case when v_date > sysdate
then
add_months(v_date, -12)
else
v_date
end,v_format);

exception when others then
raise_application_error
(-20002, 'in_date='|| pTRANSMISSION_DATE_TIME||' ERR:'||sqlerrm);
END;

FUNCTION Set_RDATE_ID (pTRANSMISSION_DATE_TIME varchar2)
RETURN NUMBER
IS
vDATE VARCHAR2(20 BYTE);
vRDATE1 DATE;
vRDATE1_string VARCHAR2(20 BYTE);
vRDATE_ID NUMBER;
vDate_CHINA Date;
vDate_UKRAINE Date;
v_format_1 varchar2(20) := 'YYYYMMDDHH24MISS';
v_format_2 varchar2(8) := 'YYYYMMDD';

BEGIN

SELECT UPI_ECL.Set_Transmission_Date_Time2
(pTRANSMISSION_DATE_TIME)
INTO vDATE from dual;

DBMS_OUTPUT.PUT_LINE('vDATE = ' || vDATE);

vDate_CHINA := TO_DATE(vDATE, v_format_1);

DBMS_OUTPUT.PUT_LINE
('vDate_CHINA = ' || TO_CHAR(vDate_CHINA, v_format_1));

vDate_UKRAINE :=
vDate_CHINA - (1/24*6); -- отнимаем 6 часов. чтобы получить украинское
время

DBMS_OUTPUT.PUT_LINE

```

```

('vDate_UKRAINE = ' || TO_CHAR(vDate_UKRAINE, v_format_1));

vRDATE1_string := TO_CHAR(vDate_UKRAINE, v_format_2);

DBMS_OUTPUT.PUT_LINE('vRDATE1_string = ' || vRDATE1_string);

vRDATE1 := TO_DATE(vRDATE1_string, v_format_2);

DBMS_OUTPUT.PUT_LINE('vRDATE1 = ' || vRDATE1);

SELECT RDATE_ID
INTO vRDATE_ID
FROM ARCDATE
WHERE RDATE1 = vRDATE1
AND PERIOD_ID = 1;

RETURN vRDATE_ID;

END;

PROCEDURE RECONCILIATION (pRdate_id number)
IS

vn_tx_unique_num number;

CURSOR MAP_TRAN_ACOMN (vRdate_id number)
is
SELECT * FROM UPI_ACOMN
WHERE RDATE_ID = vRdate_id
FOR UPDATE of tx_unique_num;

CURSOR MAP_TRAN_AC (vRdate_id number)
is
SELECT * FROM UPI_AC
WHERE RDATE_ID = vRdate_id
FOR UPDATE of tx_unique_num;

BEGIN
FOR cur_rec in MAP_TRAN_ACOMN(pRdate_id)
LOOP
select tx_unique_num
into vn_tx_unique_num
from arctrfin
where rdate_id = pRdate_id
and entry_tim = ltrim(cur_rec.order_number,'0');

```

```

UPDATE arctrfin
set DE071$SETTL_NUM = cur_rec.MSG_ID,
DED07$SETTL_DATE = to_date(SUBSTR
(cur_rec.TRANSMISSION_DATE_TIME2,1,4)
||cur_rec.SETTLEMENT_DATE,'YYYYMMDD'),

SETTL_AMT1 = cur_rec.AMOUNT_SETTLEMENT,
SETTL_CRNCY_CDE = cur_rec.CURRENCY_CODE_SETTLEMENT,
SETTL_FEE= round(cur_rec.TOTAL_FEE_AMOUNT/1000)*
Decode
(substr(cur_rec.TOTAL_FEE_DEBIT_OR_CREDIT_ID,1,1),'C',1,'D',-1),
SETTL_FILE_ID = cur_rec.FILE_ID
where tx_unique_num = vn_tx_unique_num;

```

```

UPDATE UPI_ACOMN
SET TX_UNIQUE_NUM = vn_tx_unique_num
WHERE CURRENT OF MAP_TRAN_ACOMN;

```

```

END LOOP;

```

```

FOR cur_rec in MAP_TRAN_AC(pRdate_id)
LOOP
select tx_unique_num
into vn_tx_unique_num
from arctrfin
where rdate_id = pRdate_id
and entry_tim = ltrim(cur_rec.order_number,'0');

```

```

UPDATE arctrfin
set DE071$SETTL_NUM = cur_rec.MSG_ID,
DED07$SETTL_DATE = to_char
(SUBSTR(cur_rec.TRANSMISSION_DATE_TIME2,1,4)
||cur_rec.SETTLEMENT_DATE,'DD.MM.YYYY'),
SETTL_AMT1 = cur_rec.AMOUNT_SETTLEMENT,
SETTL_CRNCY_CDE = cur_rec.CURRENCY_CODE_SETTLEMENT,
SETTL_FEE = round(cur_rec.TOTAL_FEE_AMOUNT/100),
SETTL_FILE_ID = cur_rec.FILE_ID
where tx_unique_num = vn_tx_unique_num;

```

```

UPDATE UPI_AC
SET TX_UNIQUE_NUM = vn_tx_unique_num
WHERE CURRENT OF MAP_TRAN_AC;

```

```

END LOOP;

```

```
COMMIT;
END;
```

```
FUNCTION GET_CNTRY_BY_PAN(pPan varchar2) RETURN VARCHAR2
IS
CURSOR c1 IS
SELECT
ECCF_G.Get_ALPHACNTRY_BY_NUMCNTRY(to_number(ISSUING_REGIO
N)) ALPH_COUNTRY_CDE
from upi_bin_rec
where bin = substr(pPan,1,bin_length);
```

```
vRez varchar2(3);
```

```
BEGIN
vRez := 'UNK';
FOR c1_rec IN C1 LOOP
vRez := c1_rec.ALPH_COUNTRY_CDE;
EXIT;
END LOOP;
RETURN vRez;
end;
END UPI_ECL;
/
```

ДОДАТОК Д
Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Кафедра програмних засобів

Кваліфікаційна дипломна робота
бакалавра

**на тему: РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ
АВТОМАТИЗАЦІЇ КЛІРИНГУ У МІЖНАРОДНІЙ ПЛАТІЖНІЙ
СИСТЕМІ UNIONPAY**

Виконав:

Косінов А. В.
студ. групи КНТз - 118

Керівник:

Субботін С. О.
зав. кафедри ПЗ, д.т.н., професор

Рисунок Д.1 – Титульний аркуш презентації

Об'єкт, предмет дослідження та мета роботи

- **Об'єкт дослідження** – процес розробки та функціонування модуля бази даних для роботи з кліринговою системою;
- **Предмет дослідження** – методи розробки та підтримки функціонування модуля бази даних призначеного для обробки транзакцій клірингової системи;
- **Мета роботи** – розробити модуль бази даних який автоматизує кліринг у міжнародній системі UPI.

2

Рисунок Д.2 – Об'єкт, предмет дослідження та мета роботи

Аналіз вимог

Завданням в кваліфікаційній дипломній роботі - є вилучення, завантаження та перетворення даних до необхідного формату. Реалізацію розділено на наступні етапи:

- 1) вилучення транзакцій з файлів даних;
- 2) створення тимчасових таблиць;
- 3) за допомогою контрол-файлів перетворення транзакцій до табличного вигляду в тимчасових таблицях;
- 4) створити пакет баз даних для обробки транзакцій з тимчасових таблиць і перетворення даних для подальшої інтеграції з працюючою системою.

3

Рисунок Д.3 – Аналіз вимог

Порівняння підходів вилучення, завантаження та перетворення даних

Функція	ETL	ELT
Режим обробки даних	Пакетний, близький до реального часу	Пакетний, близький до реального часу
Обробка даних	У пам'яті сервера або кластера серверів	На рівні СУБД
Вимоги	Великі вимоги до ресурсів сервера (-ів)	Великі вимоги до потужності СУБД
Приймачі даних	Розроблене під ETL програмне забезпечення	Приймачем даних звичайно є СУБД, на якій ведеться обробка даних

4

Рисунок Д.4 – Порівняння підходів ELT та ETL

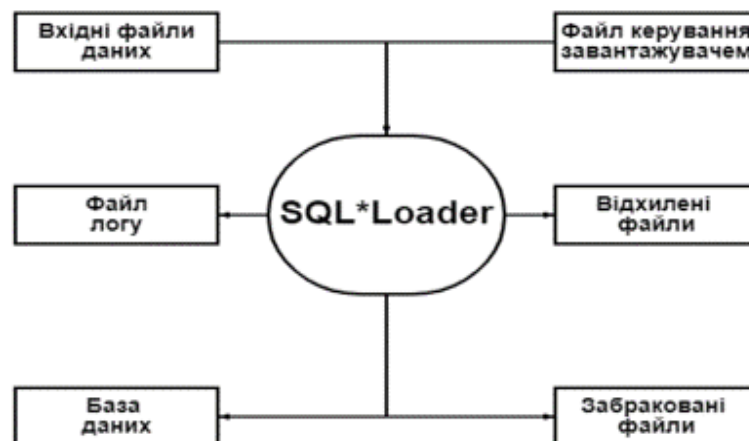
Схема технологій вилучення даних



5

Рисунок Д.5 – Схема вилучення даних

Схема перетворення вхідних даних



6

Рисунок Д.6 – Схема перетворення вхідних даних

Схема алгоритму трансформації даних



7

Рисунок Д.7 – Схема алгоритму трансформації даних

Порівняння СУБД

Характеристики	ORACLE	MS SQL SERVER
Мови програмування	Java, SQL, PLSQL	C#, Transact – SQL, Transact - PLSQL
Максимальний розмір таблиць	Не обмежено	Обмежено
Максимальне число таблиць в	Не обмежено	Не обмежено
Об’єктно-орієнтовне проектування	Так	Ні
Максимальне число користувачів	Не обмежено	Майже не обмежено
Мультимедійний тип даних	Так	Ні

8

Рисунок Д.8 – Таблиця порівняння характеристик СУБД

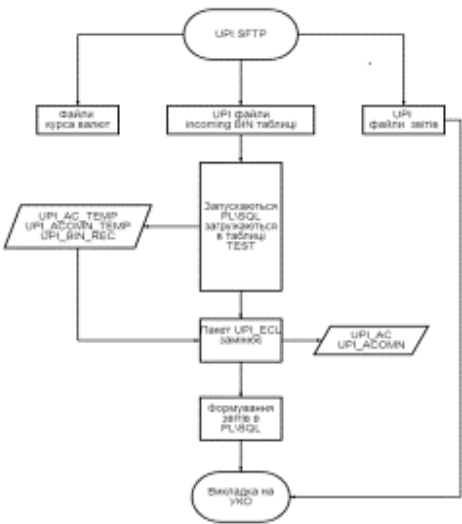
Порівняння середовищ розробки

Критерії порівняння	Середовища розробки		
	Toad	Oracle SQL Developer	SQL Database Studio
Кросплатформність	-	+	-
Підтримка декількох серверів	+	+	-
Моніторинг серверів	+	-	+
Відладчик	+	±	±
Підтримка репозиторія	+	+	+
Робота з різними СУБД	+	+	-
Доступність	+	+	±

9

Рисунок Д.9 – Таблиця порівняння характеристик середовищ розробки

Схема функціонування модулю бази даних



10

Рисунок Д.10 – Схема функціонування модулю БД

Приклад вивантаження транзакцій

The screenshot shows a Microsoft Excel spreadsheet with a large table of transaction data. The table has many columns, with headers in English and Ukrainian. The data is organized in rows, with some cells containing formulas or references. The spreadsheet is titled 'Приклад вивантаження транзакцій' (Example of transaction export).

Рисунок Д.11 – Приклад вивантаження транзакцій

Висновки

- В рамках кваліфікаційної дипломної роботи було розроблено модуль бази даних, який дозволяє автоматично обробляти транзакції міжнародної системи UPI.
- Створений модуль бази даних пройшов тестування, на сьогоднішній момент недоліків не виявлено. Він має усі необхідні функції: створення, видалення та оновлювання таблиць.
- Практична цінність роботи полягає у тому, що створений модуль бази даних перетворює транзакції до вигляду міжнародних стандартів, для подальшої обробки даних в процесингових центрах і банках, що в свою чергу заощаджує час та кошти компанії.

Рисунок Д.12 – Висновки дипломної роботи