

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ  
РОБОТИ СИСТЕМИ ОБЛІКУ КОРИСТУВАЧІВ БАСЕЙНУ**

Здобувач освіти гр. ІН.м-01н

М.І. Коренев

Науковий керівник,  
кандидат ф.-м. наук

Г.А. Олексієнко

Завідувач кафедри  
доктор технічних наук, професор.

А.С. Довбиш

Суми 2022

Сумський державний університет  
(назва вузу)

Факультет ЕліТ Кафедра Комп'ютерних наук  
Спеціальність 122 «Комп'ютерні науки»

Затверджую:  
зав.кафедрою \_\_\_\_\_  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

Кореневу Миколі Івановичу  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія підвищення  
ефективності роботи системи обліку користувачів басейну

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи) \_\_\_\_\_

3. Вхідні данні до проекту (роботи) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Інформаційний огляд 2) Вибір програмних засобів 3) Практична  
реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_ (підпис)

Завдання прийняв до виконання \_\_\_\_\_ (підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Інформаційний огляд</i>		
2.	<i>Вибір програмних засобів</i>		
3.	<i>Практична реалізація</i>		
4.	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Студент – дипломник \_\_\_\_\_ (підпис)

Керівник проекту \_\_\_\_\_ (підпис)

## РЕФЕРАТ

**Записка:** 90 стор., 30 рис., 1 табл., 2 додатки, 22 джерела.

**Об'єкт дослідження** — Інформаційна технологія підвищення ефективності роботи системи обліку користувачів басейну.

**Мета роботи** — розробка та впровадження рекомендацій підвищення ефективності роботи системи обліку користувачів басейну.

**Методи дослідження** — системно-інформаційний аналіз, інформаційне моделювання та комп'ютерний експеримент.

**Результати** — розроблені та впроваджені конкретні рекомендації щодо підвищення ефективності роботи системи обліку користувачів басейну. Зроблено висновки щодо потенціалу оптимізації інформаційної технології.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ  
РОБОТИ СИСТЕМИ ОБЛІКУ КОРИСТУВАЧІВ БАСЕЙНУ,  
ORACLE, SQL

## ЗМІСТ

ВСТУП .....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
1.1 Аналіз предметної області.....	8
1.2 Огляд методів оптимізації SQL запитів .....	14
1.3 Постановка задачі .....	23
2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ .....	26
2.1 Інтерпретатор командного рядка SQL*Plus .....	26
2.1 Середовище розробки Oracle SQL Developer .....	27
2.2 Графічний інструмент Oracle SQL Developer Data Modeler.....	32
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ .....	39
3.1 Візуалізація схеми бази даних .....	39
3.2 Базова оптимізація запитів.....	39
3.3 Поглиблена оптимізація запитів.....	43
ВИСНОВКИ.....	50
СПИСОК ЛІТЕРАТУРИ.....	51
Додаток 1. Створення додаткових індексів.....	53
Додаток 2. Фіксація статистики Oracle.....	55

## ВСТУП

Неперервний розвиток інформаційних технологій веде до постійного збільшення інформаційного навантаження, а отже підвищуються вимоги до швидкості отримання, якості обробки та зберігання інформації. Поява нових типів інформаційних систем створює потребу в необхідності ефективно зберігати та обробляти великі обсяги інформації. [1]

Для ефективної обробки та зберігання даних активно використовуються різні типи баз даних. Взагалі, під поняттям бази даних можна вважати будь-який структурований набір даних (навіть паперову картотеку). Однак розглядаючи сучасне поняття баз даних, з точки зору інформатики, мова йде лише про електронні форми зберігання даних, систему управління базами (СУБД) призначеними для систематизованого зберігання даних і можливості швидкого доступу до них з використанням різних комп'ютерних програм. Зокрема, однією з найпопулярніших форм є реляційна база даних запропонована Едвардом Коддом в 1970 році та заснована на реляційні моделі даних. Організація різних елементів даних у вигляді набору формально описаних таблиць з нормалізованими відносинами різного ступеня і становить основу реляційної бази даних. [6]

Збільшення обсягу даних веде і до збільшення часу, котрий витрачається на пошук та обробку інформації, збільшується час обробки запитів сервером. З одного боку вдосконалюючи технічну базу, збільшуючи продуктивність комп'ютерів з СУБД можна отримати значне покращення швидкості обробки запитів. Однак просто покращення швидкості обробки запитів лише шляхом вдосконалення технічної бази не є достатнім та ефективним. Для досягнення більшого ефекту можна змінами та вдосконалити алгоритми обробки даних. Розроблена безліч таких способів для збільшення швидкості виконання запитів. Так, наприклад, при використанні індексів у вигляді забалансованих дерев можна отримати значний приріст швидкості обробки даних в порівнянні

з лінійним вдосконалення технічної бази. Це далеко не єдиний спосіб. Так ще можна навести приклади з матеріалізованими та секційними уявленнями, хеш-функціями, різними алгоритмами об'єднання та обмеження даних, розподілу значень по таблицях, ведення статистики, алгоритмів побудови та вибору оптимального плану виконання запиту і багато іншого. [8]

Роботи з оптимізації запитів ведуться уже не одне десятиліття і з плином часу вони не втратили свою актуальність, а навпаки збільшення об'ємів даних та більш жорсткими вимогами до систем такі роботи стають ще більш актуальними для вирішення повсякденних задач.

Гострій проблемі в ефективній оптимізації запитів і присвячена дана робота, а саме в необхідності провести аналіз запитів для зовнішньої системи (вебресурс), виокремити з них проблемні, провести свого роду аудит БД, знайти можливі неефективності в запитах, неефективності в організації структури даних та підвищити швидкодію запитів лише на рівні БД, без необхідності будь-якої зміни в іншій частинах.

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Аналіз предметної області

До 2030 року 90 відсотків населення у віці 6 років і старше будуть онлайн і це не лише передбачення на майбутнє, а й сухі цифри досліджень проведених дослідницькою компанією Cybersecurity Ventures. [2]

Зокрема в дослідженні “Humans On The Internet Will Triple From 2015 To 2022 And Hit 6 Billion” наводяться дані про те, що до активних користувачів інтернету ввійдуть нові групи населення, такі як люди похилого віку, діти та жителі країн, що розвиваються. [3] А отже актуалізація проблем все більшого зростання обсягу інформації й пов’язаних з цим інших проблем стане неминучою. Зокрема можна виділити:

- 1) Проблема збереження великих обсягів даних. Розглядаються наступні можливі рішення:
  - Фізичне збільшення сховищ даних, але не до нескінченності.
  - Покращення схем і способів архівування та зберігання даних.
  - Виключення дублюючих і повторюваних даних.
- 2) Проблема оброблення великих обсягів даних інформаційними системами. Інтенсивні дослідження в згаданій проблематиці сфокусовані на [4][5]:
  - Покращення швидкодії обчислювальних алгоритмів.
  - Розробка гібридних архітектурних систем.
  - Створення нових способів та систем індексування даних.
  - Розробка нових моделей баз даних таких як NoSQL, NewSQ тощо.
  - Розробляються підходи до перетворення SQL запитів в структуру дерева запитів для реляційних баз даних.

Розглядаючи процес прискорення обробки даних, зокрема для реляційних баз даних, першим фундаментальним кроком, описаним ще



Едвардом Коддом є нормалізація схеми бази даних та подальша денормалізація відповідно до різних форм запиту. [6]

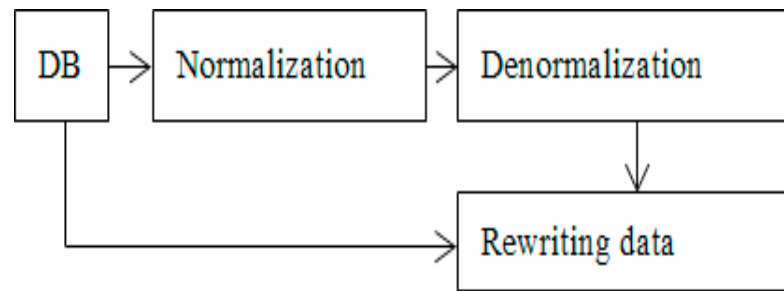


Рисунок 1.1 – Схематичне представлення модифікації схеми бази даних

Ця основа закладається ще при проектуванні та створенні БД і має базуватись на продуманих, не складних структурах, для недопущення створення надскладних логічних запитів та не містити таблиці з дуже великою кількістю стовпців, що може призводити до непередбачуваних витрат. Також проектування з чисельною кількістю таблиць та невеликим числом стовпців призводить до зростання числа об'єднань і написання великих за розміром запитів. Загальні методи нормалізації та денормалізації добре вивчені за не одне десятиліття, а тому потреби зосереджувати свою увагу на них немає потреби при по ознайомчому огляді.

Розуміючи основу правильного проектування БД і перед розглядом підходів та методів оптимізації різних запитів потрібно розглянути сам процес обробки запитів, схематично зображеного на рис. 1.2.

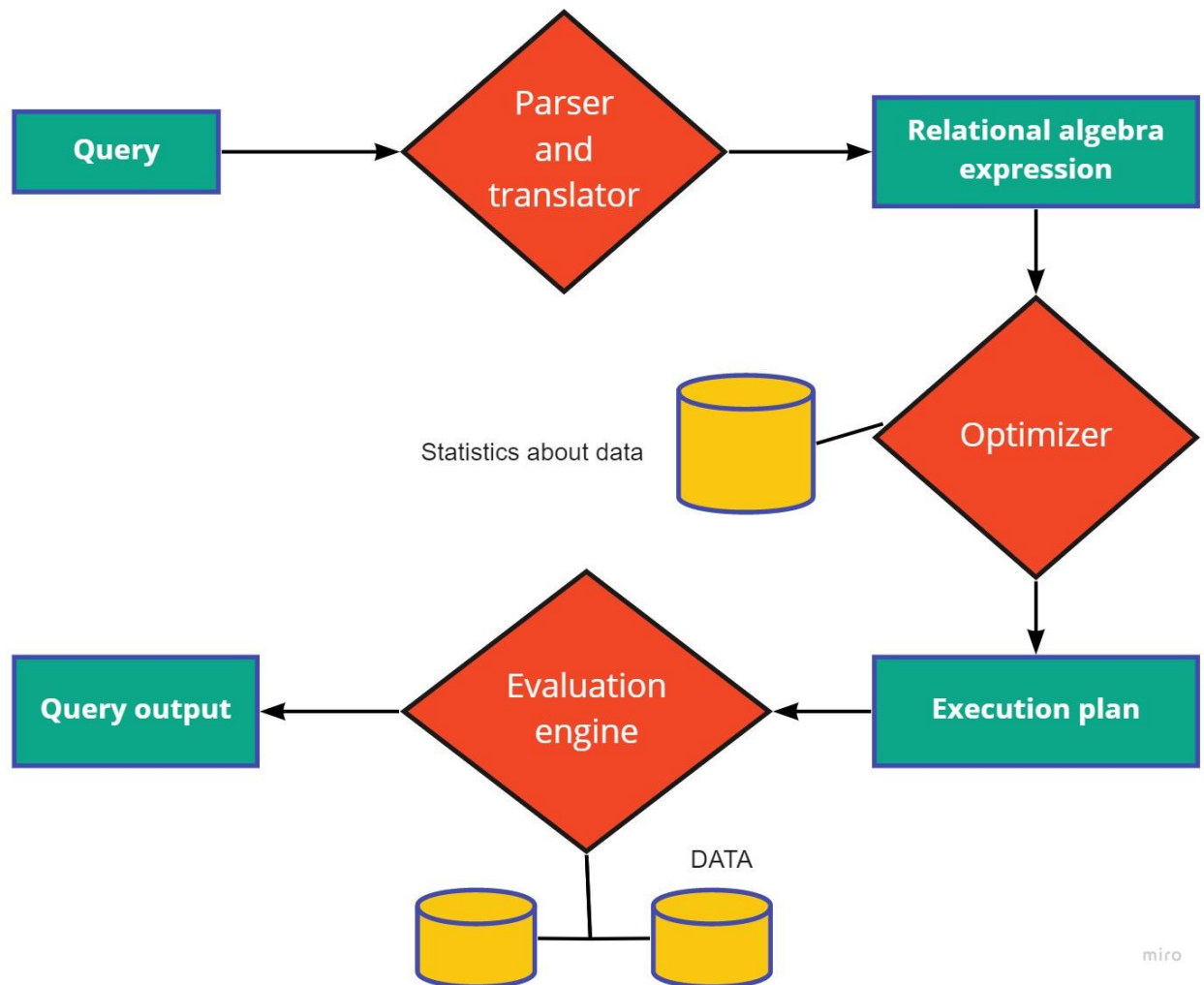


Рисунок 1.2 – Схематичне зображення обробки запиту в СУБД Oracle

Обробка самого запиту може розглядатись як група етапів пов'язаних між собою для отримання чи модифікації даних в БД.

Обробка запитів має три ключові етапи:

- 1) Parser and translator;
- 2) Optimization;
- 3) Execution plan.

На першому кроці виконання запиту (Parser and translator) відбувається перевірка синтаксису написання запиту та перевірка існування вказаних сутностей в базі даних. Також відбувається переклад написаного запиту з мови SQL на нижчий машинний рівень, з використанням реляційної алгебри. На етапі оптимізації системі потрібно зібрати системні дані про необхідні

сутності БД для запиту і в залежності від організації даних запит може мати різні відповідні вирази в реляційному відображенні. Побудова плану запиту безпосередньо базується на системних даних та інформації про ці дані та має обрати оптимальний план для отримання даних по початковому запиту. При побудові планів запиту, різні варіанти мають різну вартість виконання такого запиту, що в собі може включати кількість звернень до диска, процесорний час на виконання запиту, час зв'язку у разі розподілених баз даних та інші витрати. [7]

Лише після виконання всіх описаних вище етапів відбувається безпосереднє виконання адаптованого запиту з поверненням результату користувачу.

Схематичне зображення процесу виконання запиту на рис. 1.2 дає перші уявлення про можливі проблеми з виконанням запитів на кожному з етапів. Так проблеми можуть виникнути на етапі парсингу, при неправильному написанні запиту та на етапі оптимізації та вибору ефективного плану запиту. Важливим компонентом, на який слід звернути увагу є оптимізатор Oracle.

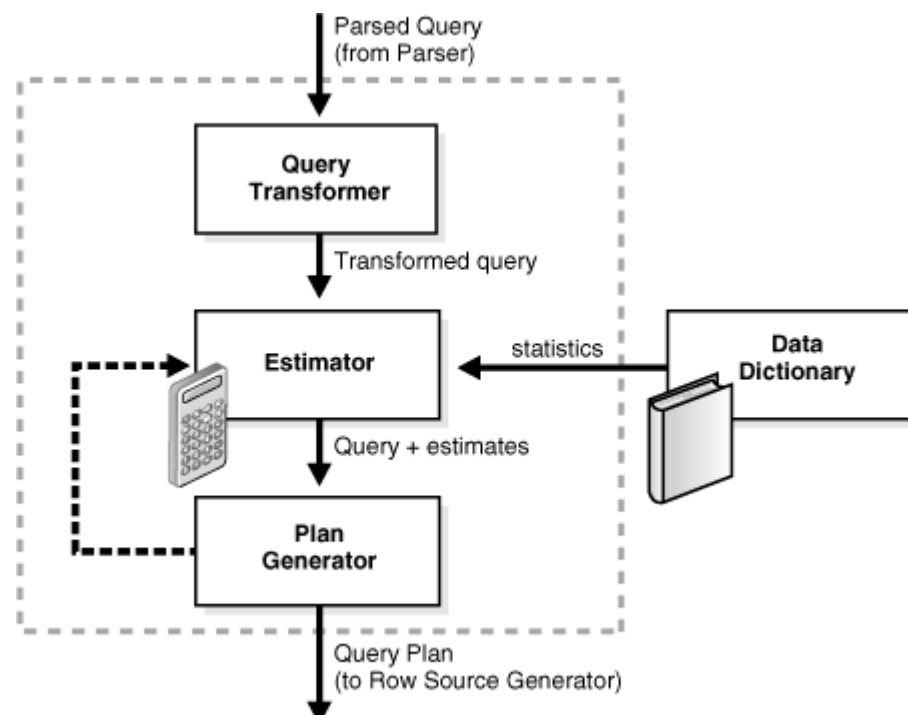


Рисунок 1.3 – Схематичне зображення компонентів оптимізатора Oracle

Складові елементи оптимізатора Oracle представлені на рис 1.3:

- 1) Query Transformer – призначений для зміни вхідного запиту за допомогою еквівалентних виразів для більш ефективного побудови плану запиту (приклад еквівалентного перетворення запиту наведено на рис. 1.4).
- 2) Estimator – оцінює вартість кожного плану запиту на основі статистичних даних зі словника (Data Dictionary).
- 3) Plan Generator – виконує вибір оптимального плану виконання запиту і передає на наступний крок генератору джерела рядків. [10][11]

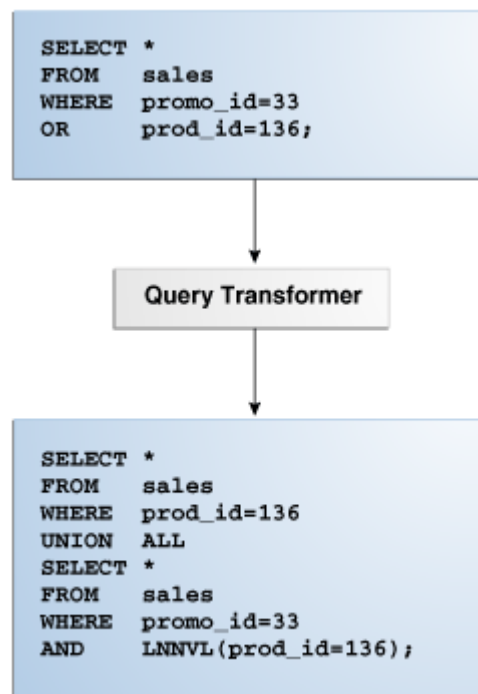


Рисунок 1.4 – Приклад еквівалентного перетворення запиту в СУБД Oracle

Коректність роботи оцінювача вартості запитів, а в результаті вибір і побудови запиту з найменшою вартістю, базується на статичних даних зі словника – статистиці. Залежно від налаштувань, статистика може бути зібраною і доступною для роботи або динамічно генеруватись. Для статистики основними ключовими параметрами є: [9]

- Selectivity - відповідає за відсоток рядків в колонці, котрі вибирає запит. Змінюється від 0 до 1 (0 – відсутні дані, 1 – всі дані колонки).

- Cardinality – кількість рядків яку повертає кожна з операцій в плані виконання запиту. Вирішальний показник при виборі плану запиту.
- Cost – показник використання ресурсів, таких як I/O, CPU, Memory usage.

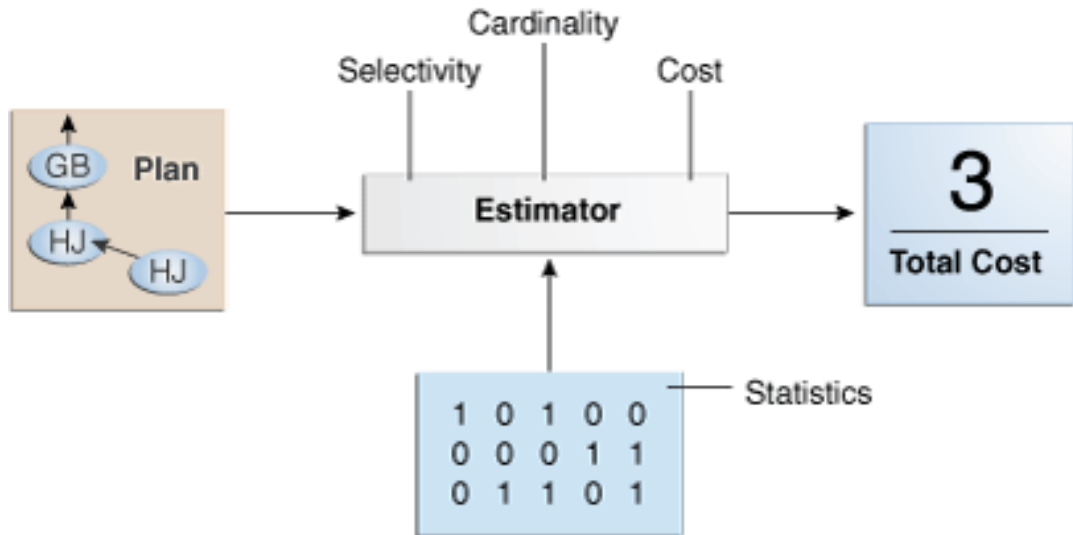


Рисунок 1.5 – Схематичне зображення процесу оцінки плану виконання запиту

Маючи базове уявлення про складові частини й призначення оптимізатора Oracle доцільно буде показати розширену схему.

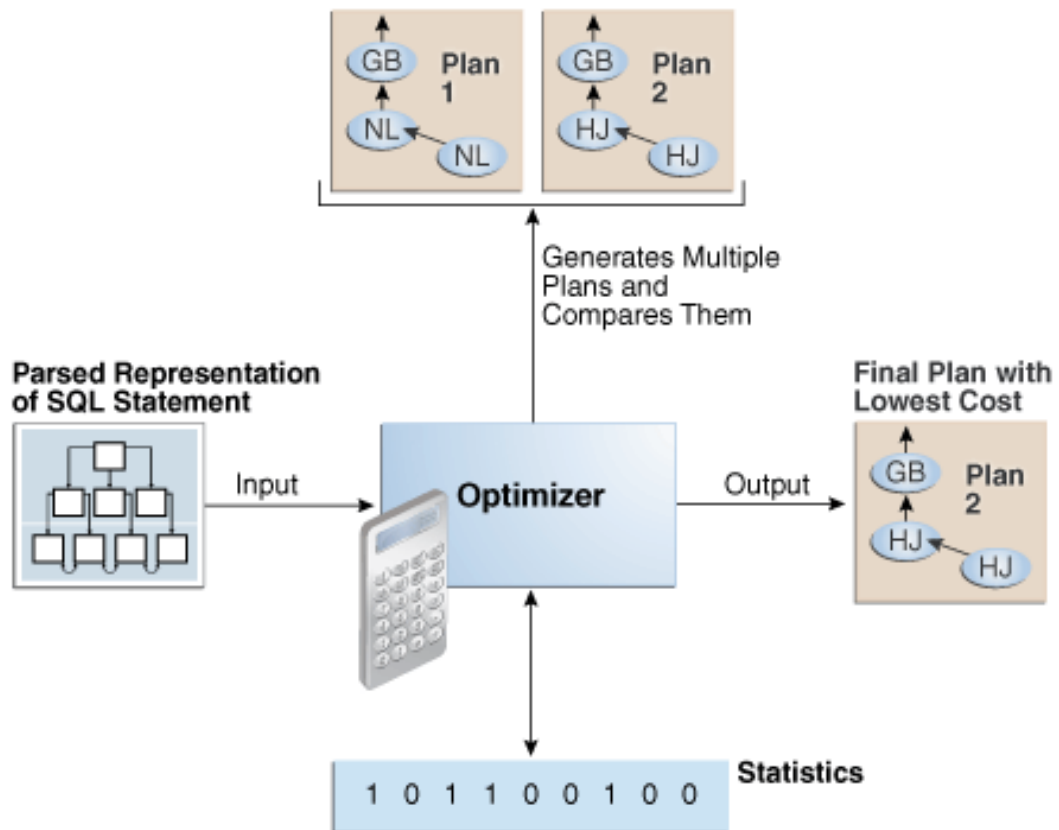


Рисунок 1.6 – Розширене схематичне зображення процесу роботи оптимізатора Oracle

Провівши аналіз проблематики та ознайомившись з основами побудови планів запитів та виконання самих запитів в реляційній базі даних Oracle, доцільно продовжити ознайомлення з підходами підвищення ефективності вибору оптимального плану виконання запиту і підвищення швидкодії виконання запиту в цілому. Підходам та методам оптимізації запитів і призначений наступний розділ роботи.

## 1.2 Огляд методів оптимізації SQL запитів

Цілі методів оптимізації SQL запитів можна поділити на 3 логічні частини:

- 1) Скорочення часу відповіді – головний показник ефективності виконання запиту розраховується як різниця в часі між користувачами, що запитують та отримують дані.

- 2) Зменшення використання процесорного часу – зменшення процесорного часу, а отже отримання швидших результатів.
- 3) Покращення пропускної здатності – зменшення числа фізичних читань та об'єму даних. Отримання, обробка та передача лише необхідної кількості рядків та стовпців. [11]

Ключовим інструментом при оптимізації SQL запитів є план запиту, процес вибору якого було розглянуто в попередньому розділі.

План запиту – послідовність операцій необхідних для отримання результату SQL запиту. Складається з двох основних частин: вибір результатів та сортування і групування цих результатів.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	1580	247 (2)	00:00:01
1	NESTED LOOPS		10	1580	247 (2)	00:00:01
2	NESTED LOOPS		10	1580	247 (2)	00:00:01
* 3	HASH JOIN		10	1320	237 (2)	00:00:01
* 4	HASH JOIN		10	1230	230 (2)	00:00:01
5	NESTED LOOPS		1	91	207 (1)	00:00:01

PLAN\_TABLE\_OUTPUT

6	NESTED LOOPS		1	91	207 (1)	00:00:01
* 7	HASH JOIN		1	30	206 (1)	00:00:01
* 8	TABLE ACCESS FULL	CLIENTS	1	10	103 (1)	00:00:01
* 9	TABLE ACCESS FULL	CLIENTS_ABONEMENTS	1658	33160	103 (1)	00:00:01
* 10	INDEX UNIQUE SCAN	XIAS_ABONEMENT_ID	1		0 (0)	00:00:01
11	TABLE ACCESS BY INDEX ROWID	ABONEMENTS	1	61	1 (0)	00:00:01
12	TABLE ACCESS FULL	GROUPS	10000	312K	22 (0)	00:00:01
13	TABLE ACCESS FULL	COACHES	998	8982	7 (0)	00:00:01
* 14	INDEX UNIQUE SCAN	XIPS_PERSON_ID	1		0 (0)	00:00:01
15	TABLE ACCESS BY INDEX ROWID	PERSONS	1	26	1 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

Predicate Information (identified by operation id):

```

3 - access("CS"."COACH_ID"="G"."COACH_ID")
4 - access("A"."ABONEMENT_ID"="G"."ABONEMENT_ID")
7 - access("C"."CLIENT_ID"="CA"."CLIENT_ID")
8 - filter("C"."PERSON_ID"=91000)
9 - filter("CA"."STATUS"='ACTIVE')
10 - access("CA"."ABONEMENT_ID"="A"."ABONEMENT_ID")
14 - access("CP"."PERSON_ID"="CS"."PERSON_ID")

```

PLAN\_TABLE\_OUTPUT

37 rows selected.

Рисунок 1.7 – Приклад плану виконання запиту

Перед безпосереднім оглядом самих методів оптимізації необхідно ознайомитись з ключовими термінами та поняттями, частину котрих можна побачити на зображеному рис. 1.7 виконання запиту.

За типами доступу до даних виділяють основні три типи:

- 1) TABLE ACCESS FULL - доступ до таблиці з повним її переглядом.
- 2) TABLE ACCESS BY INDEX ROWID - частковий доступ до таблиці за допомогою індексу.
- 3) INDEX RANGE SCAN – отримання потрібних значень з використанням сканування діапазону індексу.

За типом з'єднання вибірок даних:

- 1) HASH JOIN – для отримання результатів буде побудована хеш-таблиця.
- 2) NESTED LOOPS – отримання даних шляхом повного перегляду основної таблиці та пошуку відповідних значень в додаткові. Реалізує принцип доступу “один – до багатьох”.
- 3) SORT MERGE JOIN – використовується для об'єднання декількох джерел. Виконується сортування по спільному ключу, а потім з'єднання даних.
- 4) BUFFER SORT – при неодноразовому виконанні запиту він може бути переміщений в спеціальний блок для пришвидшення отримання даних з використанням цього запиту.
- 5) MERGE JOIN CARTESIAN – тип з'єднання з декартовим добутком. Присутність такої операції може говорити про відсутність ключа з'єднання, що може привести до несподіваних результатів. Є однією з операцій, котру потрібно уникати. [10][13][14]

Індекс – об'єкт бази даних, створюється на основі однієї або декількох колонок таблиці чи уявлення з метою підвищення продуктивності пошуку



даних. Використання індексів дозволяє знаходити рядок із певним значенням стовпця, переглядаючи лише невелику частину від загального обсягу таблиці.

Швидкодія індексів досягається внаслідок структури, оптимізованої для швидкого пошуку в порівнянні з повним скануванням таблиці при відсутності індексу. З усім тим, застосування індексів є свого роду компромісом, отримуючи прискорення запитів відбувається здорожчання операцій додавання, модифікації та видалення даних з таблиці, адже окрім даних в таблиці потрібно оновлювати й всі залежні індекси. Також створення індексів займає додаткову пам'ять, а їх розмір може перевищувати розмір початкової таблиці, адже обмежень на кількість індексів, які можуть бути створені на одній таблиці в Oracle не існує.

Індекси мають різні типи, основні з них перераховано нижче:

- Унікальні та неунікальні індекси. Унікальні індекси створюються по унікальному стовпці, а також при додаванні обмеження унікальності Oracle автоматично створює такі індекси.
- Первинні та вторинні індекси. Первинні індекси повинні мати унікальне значення, відмінне від null, автоматично створюються на первинних ключах в Oracle. Вторинні індекси – всі інші індекси, котрі можуть бути не унікальними.
- Складові індекси – індекси створені на основі декількох колонок однієї й тієї ж таблиці. Особливо корисні для забезпечення унікальності даних при комбінації декількох колонок.[15]

Маючи достатньо потужний інструмент підвищення продуктивності виконання таких як індекс, потрібно правильно їх використовувати. Надмірне, непотрібне додавання індексів може значною мірою сповільнити виконання запитів. Загальні рекомендації до використання індексів наступні:

- Доцільно використовувати індекс при одночасному доступі до 10-15% даних. Альтернативною операцією є виконання повного читання

таблиці, від початку до кінця. Використання індексу також вимагає двох операцій: читання даних з індексу, а потім читання даних за посиланням з цього індексу в таблиці.

- Доцільно створювати індекси на колонки таблиць для конструкцій where, order by, join, group by.
- Таблиці з невеликою кількістю даних не варто навантажувати індексами. Повне сканування може бути набагато кращим рішенням.
- В ідеалі, на стовпці котрі активно модифікуються повинна бути мінімальна кількість індексів, що пов'язано з накладними витратами на модифікацію індексів.
- Стовпці з великою кількістю символів в рядку – погані кандидати на індексацію.
- При створенні складових індексів першим стовпцем ключа повинний бути стовпець з мінімальною кількістю не унікальних даних. [12] [15]

Індекс таблиці повинен базуватись на типах запитів до таблиці та визначатись ними в потребі створення чи ні такого індексу.

Oracle має достатньо велику кількість різних типів індексів: B-Tree індекс, бітові індекси, індекси з реверсивним ключем, індекси зі зжатым ключем, індекси на основі функцій тощо.

B-Tree індекс – найбільш популярний тип індексів побудований як збалансоване дерево, використовується в Oracle за замовчуванням.

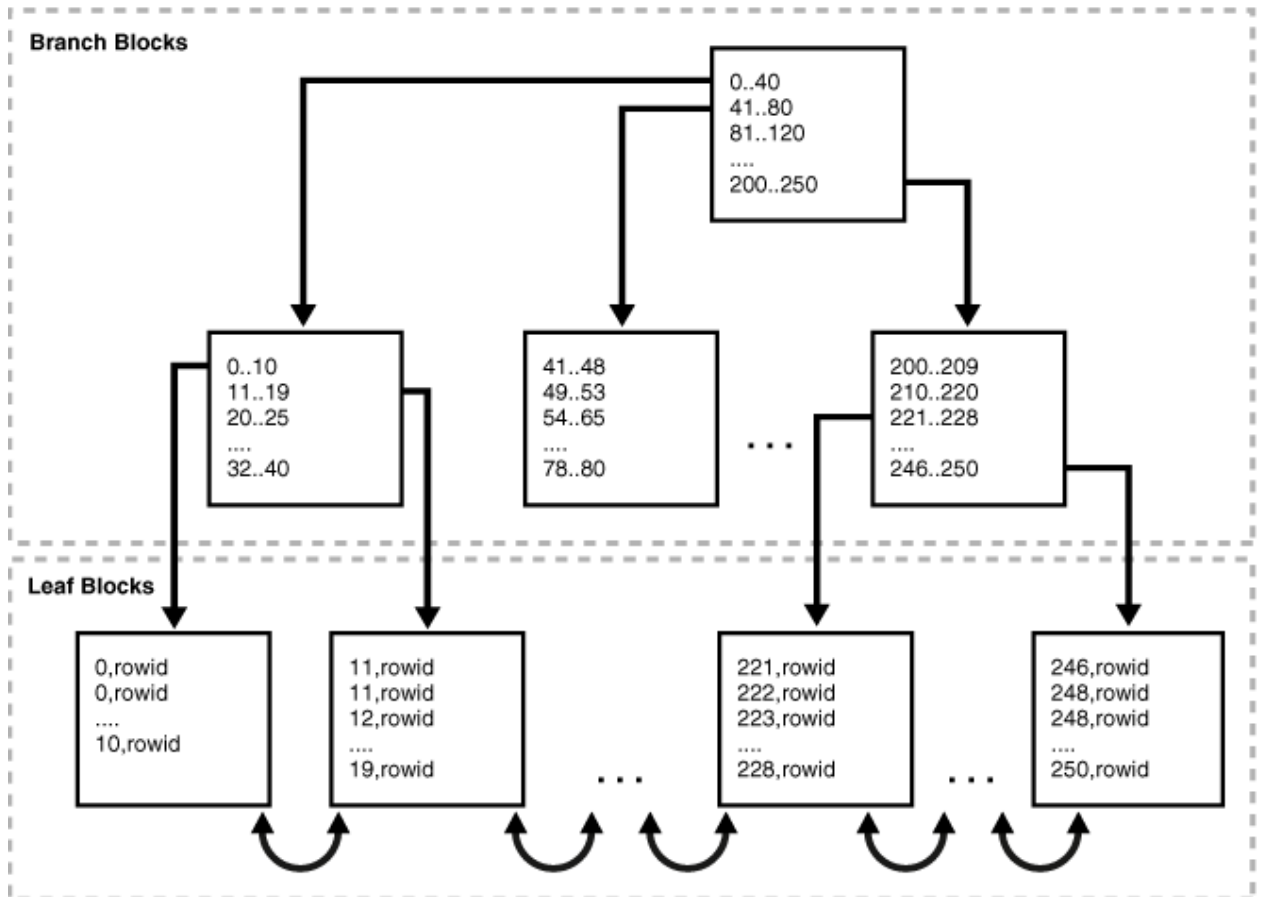


Рисунок 1.8 – Структура B-Tree індексу

Індекс збалансованого дерева представлений у вигляді зворотного дерева з однакою кількістю рівнів від будь-якого з листових блоків до кореневого вузла. Блоки верхнього рівня – блоки гілок, а блоки нижчого рівня – листя. Всі вузли, окрім вершини, мають батьківський вузол і можуть мати декілька дочірніх вузлів. Збалансовані дерева підтримують операції вибору, додавання та видалення з невеликим числом операцій вводу-виведення для кожного з операторів. При роботі з даними з B-дерева потрібно читати тільки блоки цього дерева, кількість операцій вводу-виводу буде обмежена кількістю блоків дерева та додатковими операціями читання з таблиці чи модифікацією даних в таблиці. [15]

В реалізації Oracle B-Tree індексу дерево завжди зберігає збалансований стан, з кінцевими блоками на одній глибині. Таким чином доступ до будь-якого запису займає приблизно однаковий час.

Листові блоки в собі зберігають індексовані значення стовпця та ідентифікатор рядка, який містить ці значення (ROWID). Швидке сканування індексу дає змогу отримати безпосередній доступ до відповідного рядка з використанням ROWID.

При роботі з B-Tree індексом СУБД в пошуках потрібного значення знайде це значення за  $n$  операцій ввода-виводу, де  $n$  – висота цього індексу. Також доступні різні режими сканування індексу, зокрема:

- Full Index Scan – повне сканування індексу.
- Fast Full Index Scan – повне сканування індексу опираючись лише на дані в самому індексі, читання виконується в довільному порядку.
- Index Range Scan – впорядковане читання індексу при чітких обмеженнях вибірки.
- Index Unique Scan – сканування індексу, котре завершується при отриманні першого унікального значення.
- Index Skip Scan – сканування з пропуском початкового стовпця в складеному індексі.

Навіть після створення необхідних індексів їх використання не гарантоване в Oracle. Однак існують механізми, підказки або хінти, котрі здатні вплинути на оптимізатор при складенні плану виконання запиту. Підказки надають можливість приймати рішення розробником замість тих, котрі автоматично приймаються на рівні оптимізатора. Це дає змогу створити та використати більш ефективний план запит ніж автоматично згенерований оптимізатором.

```
SELECT /*+INDEX(XIPS_PERSON_ID) */ FIRST_NAME, LAST_NAME
FROM POOLAPP.PERSONS
WHERE PERSON_ID = 10
```

Рисунок 1.9 – Приклад запиту з використанням підказки INDEX

Існують велика кількість підказок котрі діляться на наступні загальні класифікації:

- Single-table – підказки для однієї таблиці чи уявлення такі як INDEX, USE\_NL;
- Multi-table – підказки для декількох таблиць, наприклад LEADING.
- Query block – підказки для окремих блоків запиту, наприклад STAR\_TRANSFORMATION та UNNEST;
- Statement - підказки для всього оператора SQL, наприклад ALL\_ROWS.

Серед прикладів найбільш вживаних підказок можна навести наступні:

- INDEX – підказка на використання індексу;
- USE\_NL – підказка на використання NESTED LOOPS при об'єднанні результатів вибірки;
- USE\_HASH – підказка на використання HASH JOIN при об'єднанні результатів вибірки;
- LEADING – підказка на зміну порядку доступу до таблиць;
- FIRST\_ROWS – підказка для найшвидшого отримання першого рядка з даними;
- MATERIALIZE – підказка на динамічне створення тимчасової матеріалізованої таблиці. [16]

Для більш ефективного виконання оптимізації також використовується комбінація декількох підказок.

Побудувавши необхідні індекси та застосувавши підказки можна отримати більш ефективний бажаний план виконання запиту, але такий план може бути не стабільний і змінюватись при різному наборі даних. Ключове поняття, розглянуте в розділі 1.2, для побудови оптимізатором ефективного плану запиту – статистика.

Статистика оптимізатора - це певний набір даних, що описує детальніше інформацію про базу даних та об'єкти в ній. Ці дані використовуються оптимізатором для вибору кращого плану виконання запиту для кожного з операторів SQL. Статистика містить наступні дані:

- Статистику таблиць – кількість рядків, блоків та середня довжина рядків.
- Статистика колонок – число унікальних значень в колонці, число нульових значень та розподіл даних.
- Статистика індексів – кількість листових блоків, рівні та фактори кластеризації.
- Системна статистика - I/O та CPU використання і продуктивність.[17]

Статистика оптимізатора зберігається в словнику даних.

Рекомендований підхід - дозволити Oracle автоматично збирати статистику. Автоматичне регулярно заплановане обслуговування усуває багато ручних завдань пов'язаних з оптимізатором запитів і усуває можливості неефективної побудови плану запиту через застарілі або відсутні дані. Автоматичний збір статистики ввімкнено за замовчуванням. Проте автоматичний збір статистики може бути неефективний при нестабільних таблицях, котрі змінюються з частотою більшою ніж відбувається автоматичне оновлення статистики, а також для об'єктів котрі можуть збільшити свій розмір більше ніж на 10% (в більшості випадів такі таблиці використовуються для завантаження даних). Для таблиць з масовим завантаженням даних, збір статистики повинен відбуватись зразу, після їх заповнення. [18]

При всій простоті й універсальності автоматичного збору статистики – на практиці можуть виникати непередбачувані зміни в планах виконання запитів, створені та оптимізовані за допомогою підказок запити перестають продуктивно працювати, а плани запиту мають різну структуру після зміни

певної частини даних в таблицях і індексах. Oracle дозволяє вимкнути автоматичний збір статистики й перекласти відповідальність за підтримання актуальності статистики на адміністратора бази даних.

Системний пакет DBMS\_STATS дозволяє збирати статистику за таблицями, індексами та окремими стовпцями таблиць в ручному режимі. Пакет DBMS\_STATS може використовуватись як для збору статистики за окремими таблицями, так і по всьому кластері. Під час створення нової статистики, Oracle оновлює дані в словнику на останні актуальні дані й робить недійсними всі раніше проаналізовані плани запитів. Новий план запиту буде побудовано при новому зверненні до об'єкта бази даних відповідно до оновленої статистики. Також за допомогою пакета DBMS\_STATS можна генерувати синтетичну користувачку статистику, зберігати її в словнику або в окремі таблиці в користувачській схемі. [18]

Саме підхід з використанням користувачкої статистики, з мінімальним числом її зборів, а отже і з підтриманням постійного, не змінного плану може бути використано як один зі способів оптимізації. Основна складність такого підходу полягає в правильності генерації синтетичної статистики та не допущення значного відхилення між реальними даними та створеною статистикою.

### **1.3 Постановка задачі**

Метою дипломної роботи є розробка та впровадження рекомендацій підвищення ефективності роботи системи обліку користувачів басейну. Оптимізація буде виконуватись для вебресурсу з обліку користувачів басейну, котрий розроблявся в рамках бакалаврської роботи. [1]

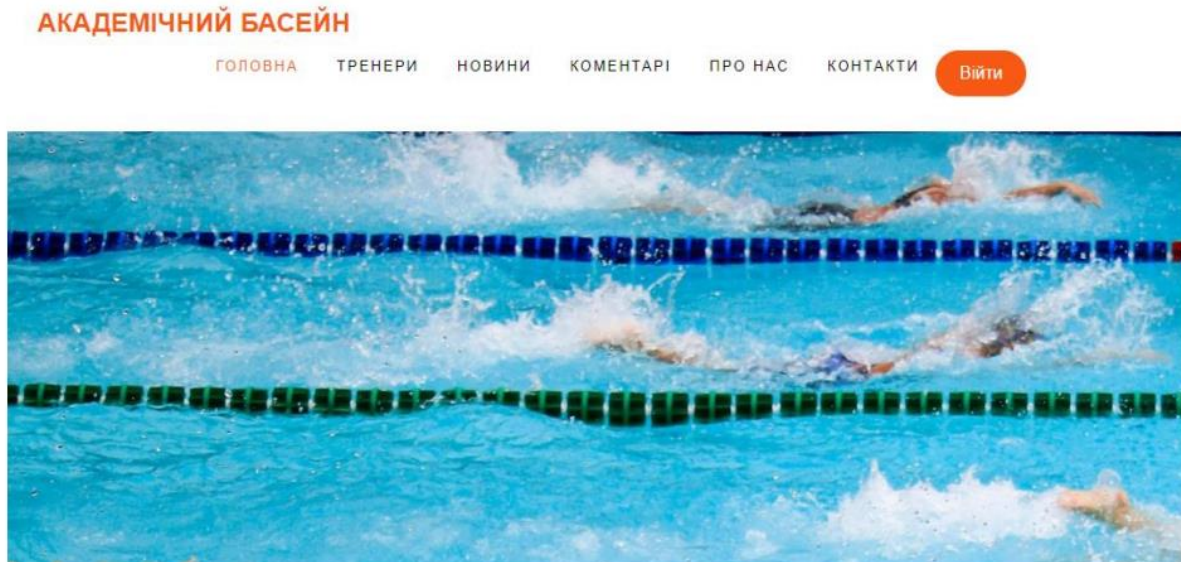


Рисунок 1.10 – Знімок початкової сторінки ІС обліку користувачів басейну

Оптимізація системи буде проводитись лише для БД (СУБД Oracle), без будь-яких змін в інших частинах додатку.

Для досягнення поставленої мети необхідно реалізувати наступні підзадачі:

- Аудит БД, її структури та зображення результатів у вигляді ERD діаграми.
- Аналіз продуктивності виконання присутніх запитів та виокремлення, по можливості, загальних проблем неефективності запитів.
- Створення додаткових індексів та таблиць та інших структур.
- Перевірка та за потреби збір і фіксація статистики ORACLE для максимально архітектурно закладеної кількості даних.
- Перевірка та аналіз продуктивності запитів з новими додатковими структурами та виокремлення проблемних запитів.
- Оптимізація виокремлених проблемних запитів за допомогою підказок (хінтів) та зі зміною структури початкового SQL запиту



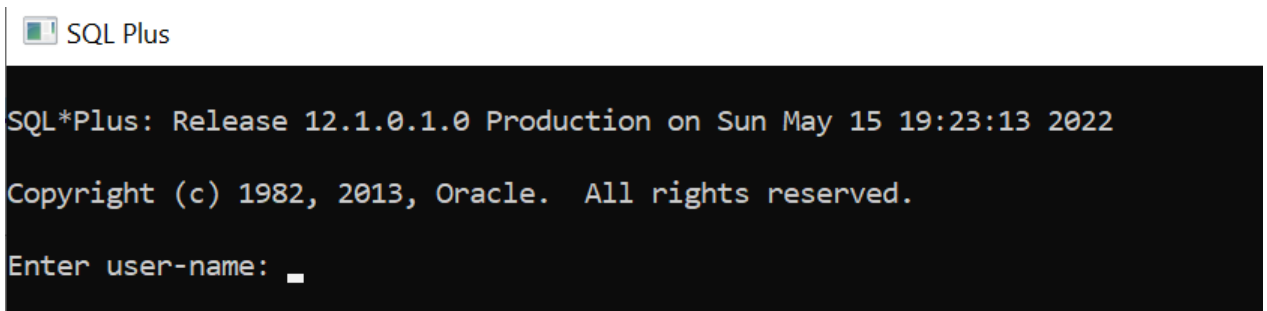
- Фінальна перевірка продуктивності запитів після виконання всіх оптимізацій.

За результатами аудиту та оптимізації скласти звіт про проведену оптимізацію та зміни внесені до БД.

## 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ

### 2.1 Інтерпретатор командного рядка SQL\*Plus

SQL\*Plus – програма-інтерпретатор яка встановлюється з установкою Oracle Database Server або Oracle Database Client. Має користувацький інтерфейс командного рядка та графічний інтерфейс Windows (GUI). Призначена для виконання команди SQL та PL/SQL в інтерактивному вигляді або зі сценарію. Активно використовується адміністраторами баз даних, розробниками та інколи навіть кінцевими користувачами. [19]



```
SQL Plus
SQL*Plus: Release 12.1.0.1.0 Production on Sun May 15 19:23:13 2022
Copyright (c) 1982, 2013, Oracle. All rights reserved.
Enter user-name: _
```

Рисунок 2.1 – Знімок вікна з логіном SQL\*Plus

SQL\*Plus початково був розроблений як спосіб введення запитів і перегляду отриманих результатів, а пізніше уже був розширений можливостями різних сценаріїв та форматування. Доволі простий базовий функціонал дає змогу виконати наступні команди:

- Виконання SELECT запитів і перегляд результатів;
- Додавання, оновлення даних в базі даних (DML операції);
- Операції DDL: створення, зміна та видалення об'єктів бази даних (наприклад, при роботі з індексами, таблицями, користувачами та їх сесіями та правами та багато іншого);
- Виконання файлових сценаріїв SQL\*Plus;
- Виконання процедур і функцій;
- Виконання PL/SQL скриптів;

- Також присутня можливість створення та форматування звітів у вигляді пакетних процесів і з можливістю виведення результатів у файл. [20]

Всі наведені приклади операцій є складовими більш скриптів та запитів, що дають змогу виконувати й отримувати різну інформацію та по різному маніпулювати з нею.

Попри значні переваги й напевно найбільша з них це можливість працювати “з коробки” без додаткової установки, SQL\*Plus не позбавлений значної кількості мінусів, котрі незбіжно відчуються при роботі. Так до прикладів таких мінусів можна віднести неможливість зручно навігуватись при виконанні багатострокового скрипту і постійна необхідність обмежувати кількість рядків, котрі потрібно отримати в виводі. Так, наприклад, при написанні наступного запиту `SELECT * FROM some_table` буде виведено всі рядки таблиці без жодних обмежень. При достатньо великій кількості таких рядків процес виведення може зайняти значну частину часу, хоча в більшості випадків метою виведення всіх записів з таблиці є не отримання всіх рядків, а отримання п кількості рядків для аналізу. Тобто потрібно писати запит з обмеженням по типу `SELECT * FROM some_table WHERE ROWNUM < 10`, що дасть змогу дістати лише 9, випадково вибраних рядків, без необхідності виведення всієї таблиці.

Маючи значну кількість переваг і недоліків, SQL\*Plus залишається базовим інструментом при роботі з базою даних. Приклади інших, більш досконалих графічних середовищ розробки наведено в наступному розділі.

## **2.1 Середовище розробки Oracle SQL Developer**

Oracle SQL Developer — це інтегроване середовище розробки на мовах SQL та PL/SQL, покликане спростити розробку та керування базою даних Oracle. Середовище розповсюджується компанією Oracle безкоштовно і

доступне для роботи на різних платформах таких як Windows, Linux і Mac OS внаслідок використання мови Java при розробці Oracle SQL Developer, що і забезпечує кросс-платформеність. Підтримка декількох платформ значна перевага, а отже при розробці сам клієнт середовища може бути встановлений на сервер з базою даних і використовуватись розробником віддалено. [21]

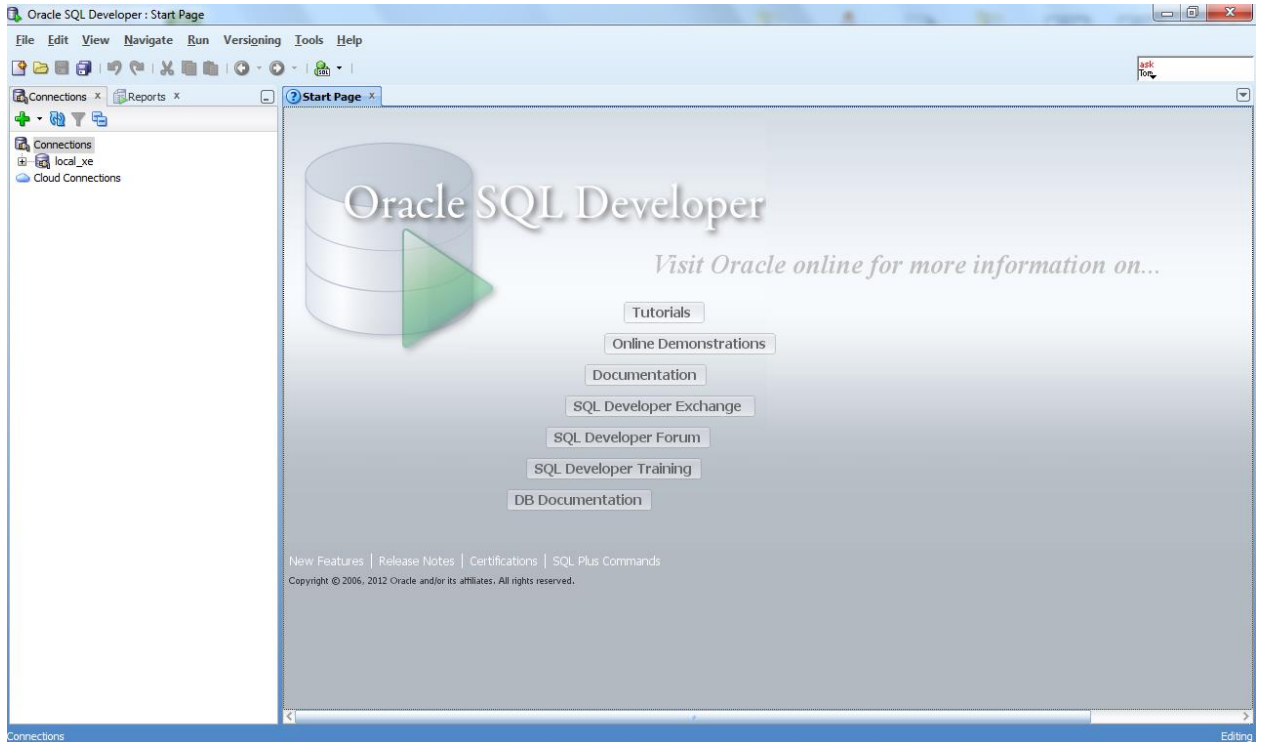


Рисунок 2.2 – Знімок вікна стартової сторінки Oracle SQL Developer

Oracle SQL Developer має зручний робочий лист з численною кількістю інструментів для виконання різних типів запитів та сценаріїв. Користувачі можуть зручно переглядати різні об'єкти бази даних, запускати різні SQL та PL/SQL запити, а також виконувати генерацію різних типів звітів. Зручна консоль значно спрощує задачі адміністрування і керування базою даних.

За замовчуванням, підключення до бази даних здійснюється через драйвер JDBC Thin. Використовуючи Oracle SQL Developer користувачі можуть під'єднатися до будь-якого випуску підтримуваної бази даних Oracle.

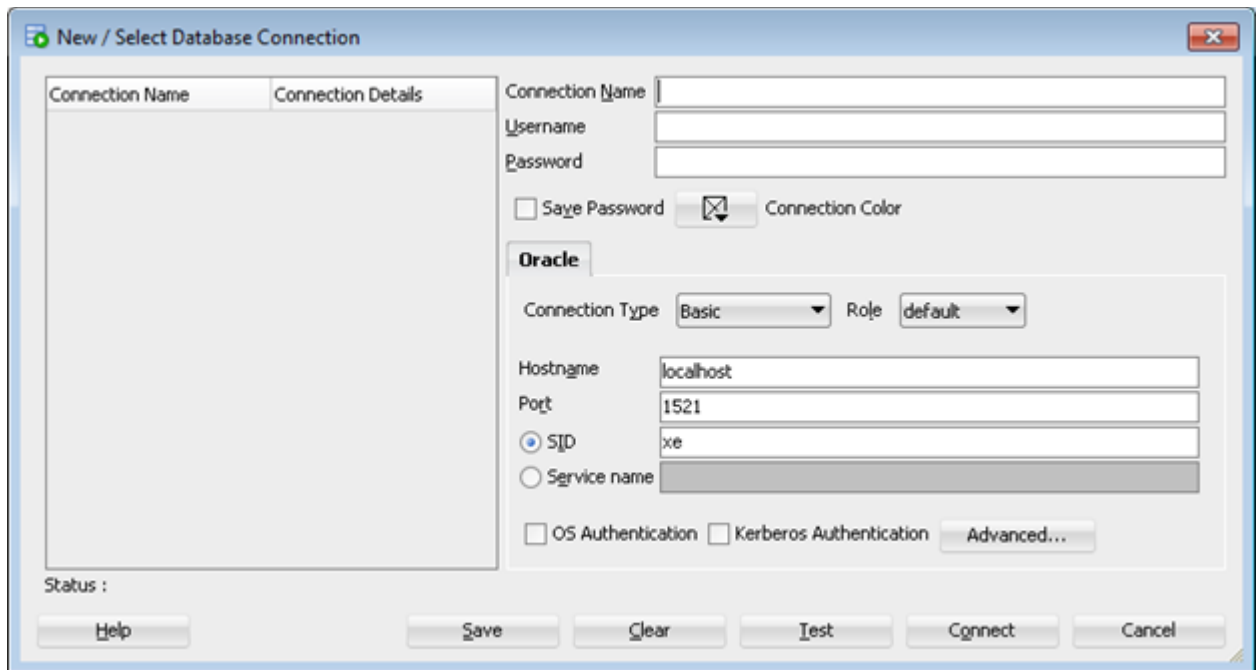


Рисунок 2.3 - Знімок вікна створення підключення до бази даних  
в Oracle SQL Developer

Розглядаючи більш детально параметри підключення зображеного на рис. 2.3 можна виділити головні пункти:

- Connection name – поле з іменем створюваного підключення.
- Username та Password – ім'я користувача та пароль для запису в базі даних Oracle до якого необхідно під'єднатися.
- Hostname – ім'я хосту чи IP-адреса сервера, на якому працює база даних Oracle.
- Port – порт для під'єднання (за замовчуванням 1521).
- SID – системний ідентифікатор Oracle (за замовчуванням XE), але може бути відсутній і необхідно буде ввести Service name.
- Service name – ім'я служби для екземпляра бази даних.

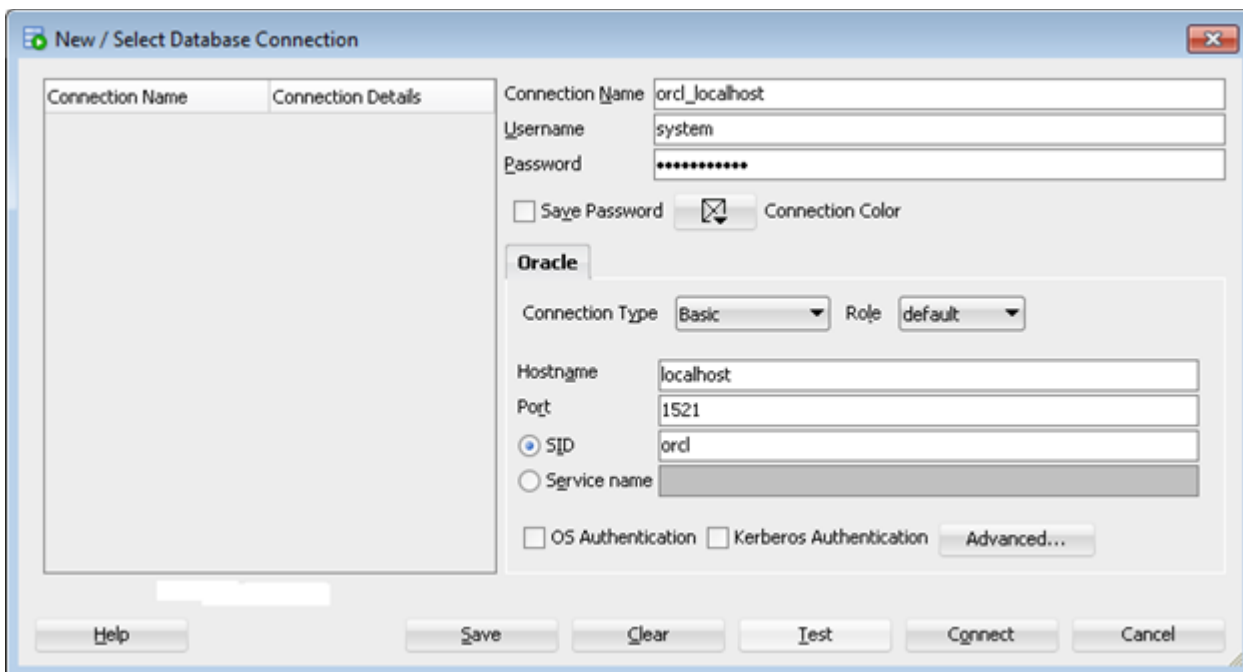


Рисунок 2.4 – Приклад підключення на локальному сервері

Після введення всіх обов'язкових базових полів можна попередньо перевірити з'єднання за базою даних нажавши на кнопку Test.

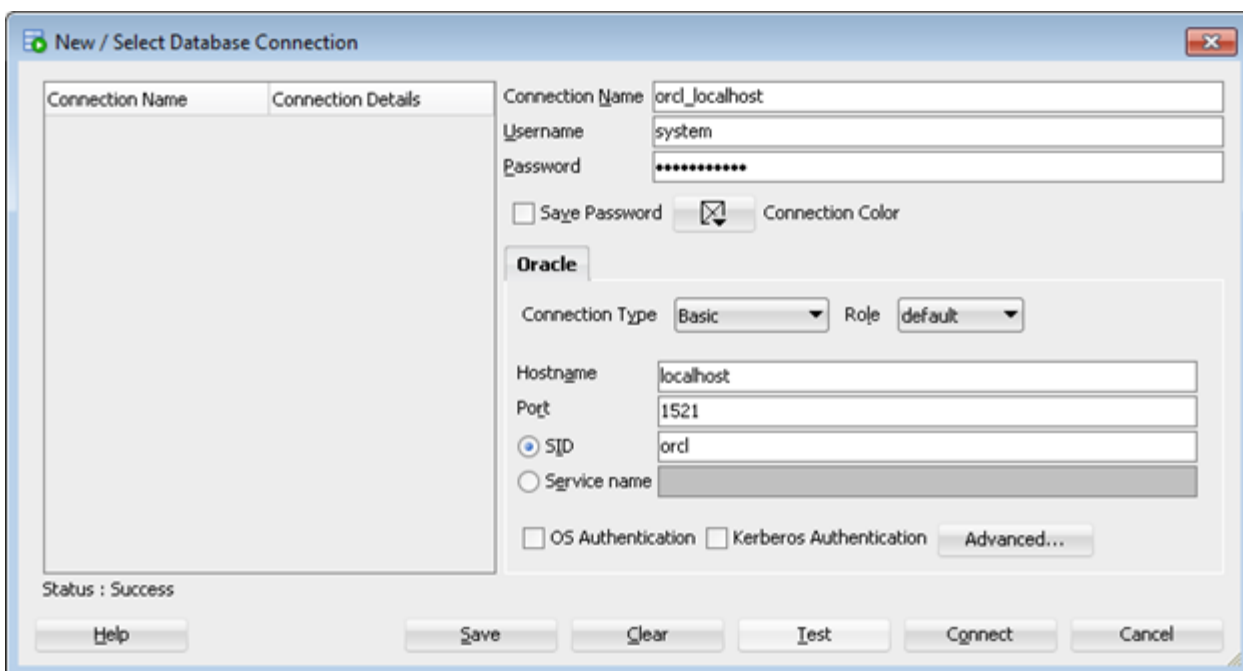


Рисунок 2.5 – Перевірка підключення до бази даних

При успішному тестовому з'єднанні можна побачити повідомлення "Status: Success" інакше висвітиться рядок з повідомлення про неможливість

підключення, наприклад при некоректному введенні ім'я користувача чи пароля можна отримати наступне повідомлення:

```
Status : Failure -Test failed: ORA-01017: invalid username/password;  
logon denied.
```

Після вирішення всіх проблем з підключення і успішному з'єднанні стартовою сторінкою буде робочий аркуш зображений на рис 2.6.

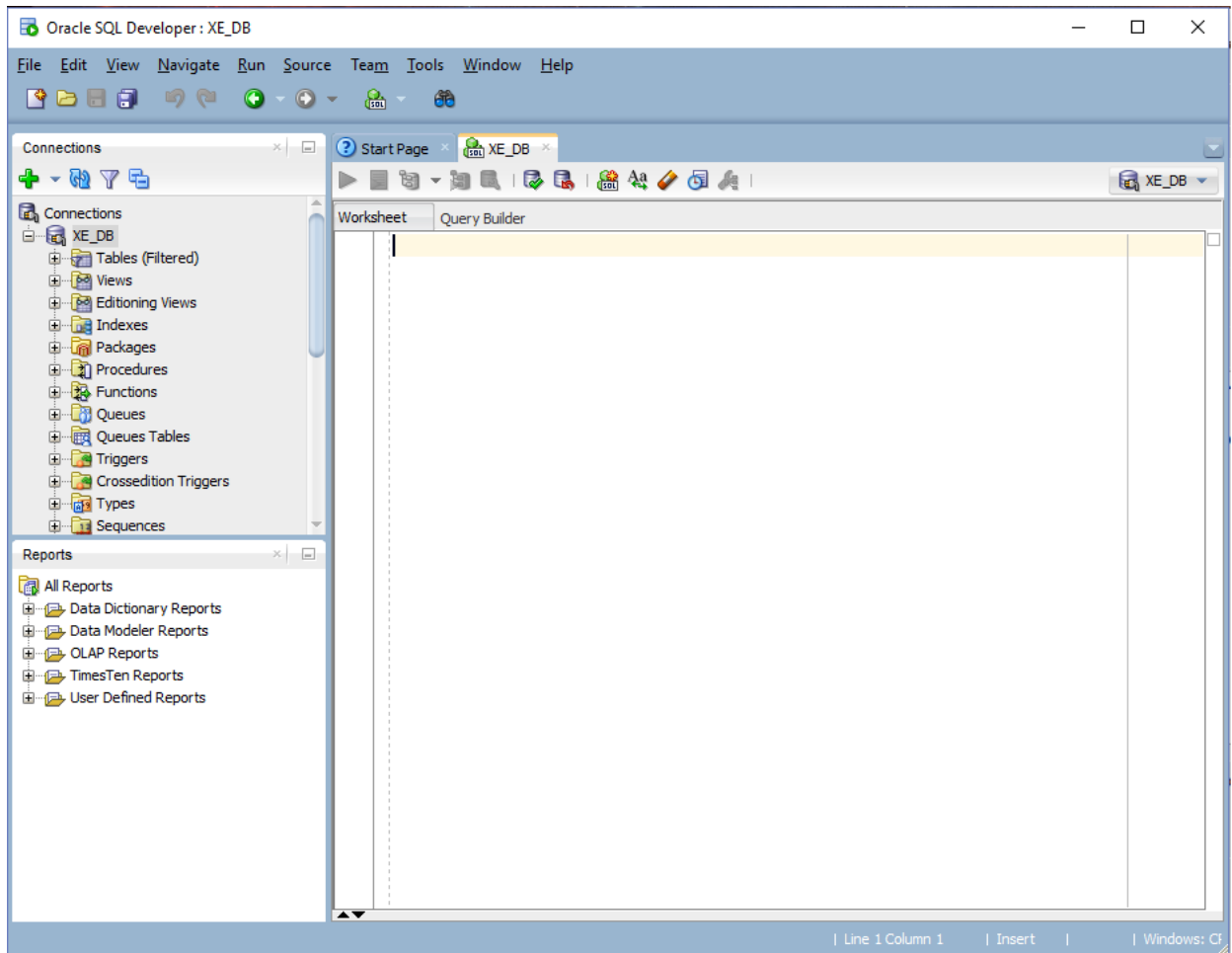


Рисунок 2.6 – Початковий робочий аркуш (worksheet) після встановлення успішного з'єднання з базою даних

Oracle SQL Developer – зручний додаток для виконання всіх необхідних маніпуляцій з даними та з підтриманням життєдіяльності бази даних взагалі. Окрім базової версії також існує і розширена версія орієнтована на візуальне моделювання та генерацію різних схем даних, що і буде розглянуто в наступному розділі.

## 2.2 Графічний інструмент Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler – безкоштовний графічний інструмент, який спрощує моделювання даних та підвищує продуктивність. Oracle SQL Developer Data Modeler може бути включений в розширену версію Oracle SQL Developer, а також може існувати як цілком незалежний додаток. Для користувачів Data Modeler надає можливості переглядати, створювати, редагувати реляційні, логічні, багатовимірні та фізичні моделі типу даних. Має підтримку прямого та зворотнього проєктування з можливостями спільної розробки. [22]

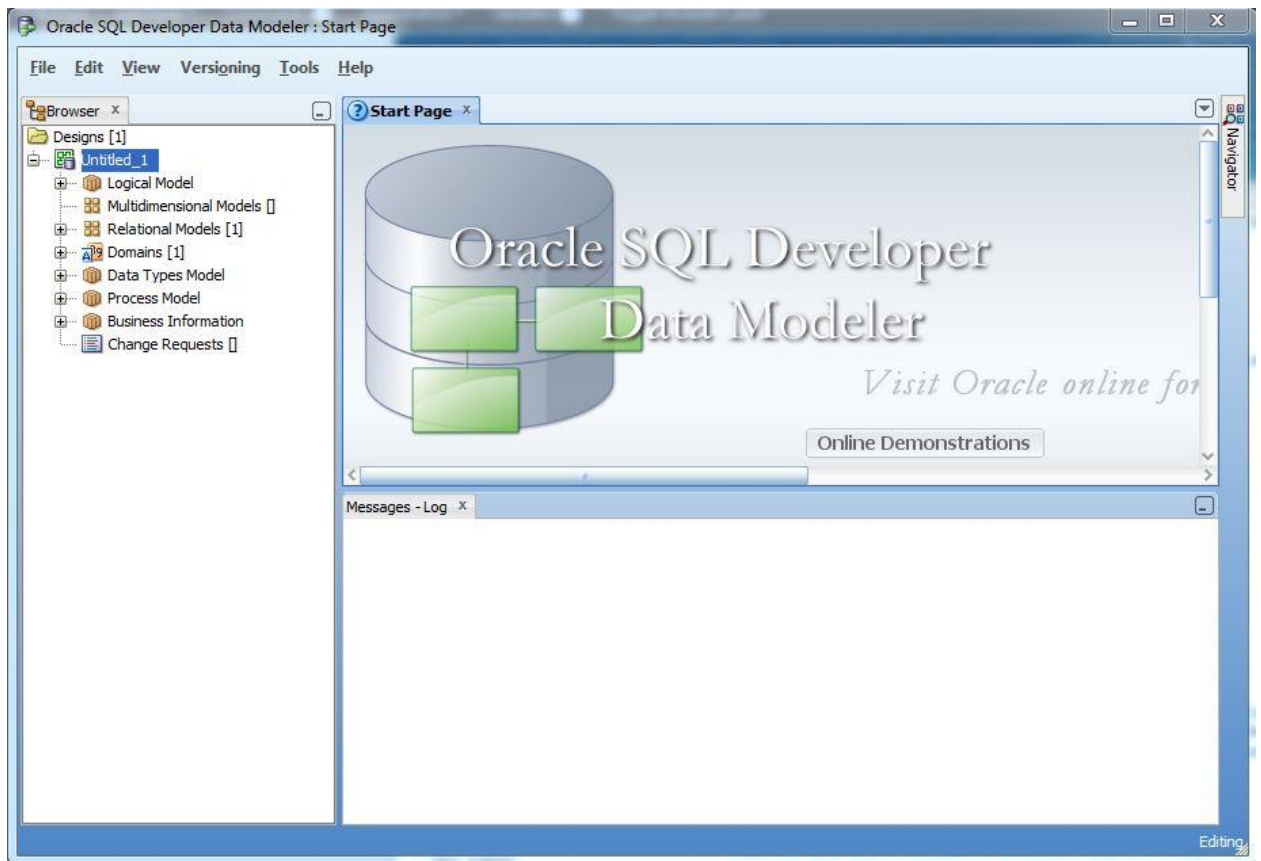


Рисунок 2.7 – Стартова сторінка Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler надає можливості для роботи з різними типами діаграм:

- ER Діаграма;
- Діаграма багатовимірних моделей;



- Діаграма типів даних;
- Реляційна діаграма;
- Діаграма потоку даних: примітивні, складені, трансформаційні завдання.

А також присутні інші різні інструменти для оптимальної роботи з візуальним відображенням даних, а саме:

- Імпорт із різних баз даних;
- Експорт для різних баз даних;
- Генерація сценаріїв DDL;
- Підставлення правил для перевірки стандартів і повноти;
- Використання шаблонів назв, глосарій і файлів скорочень для підтримки стандартів імен.

На прикладі автоматичного генерування ER діаграми для тестової схеми бази даних можна детальніше простежити особливості роботи з Data Modeler. Так початковим кроком буде імпорт словника даних з БД, що і зображено на рис. 2.8.

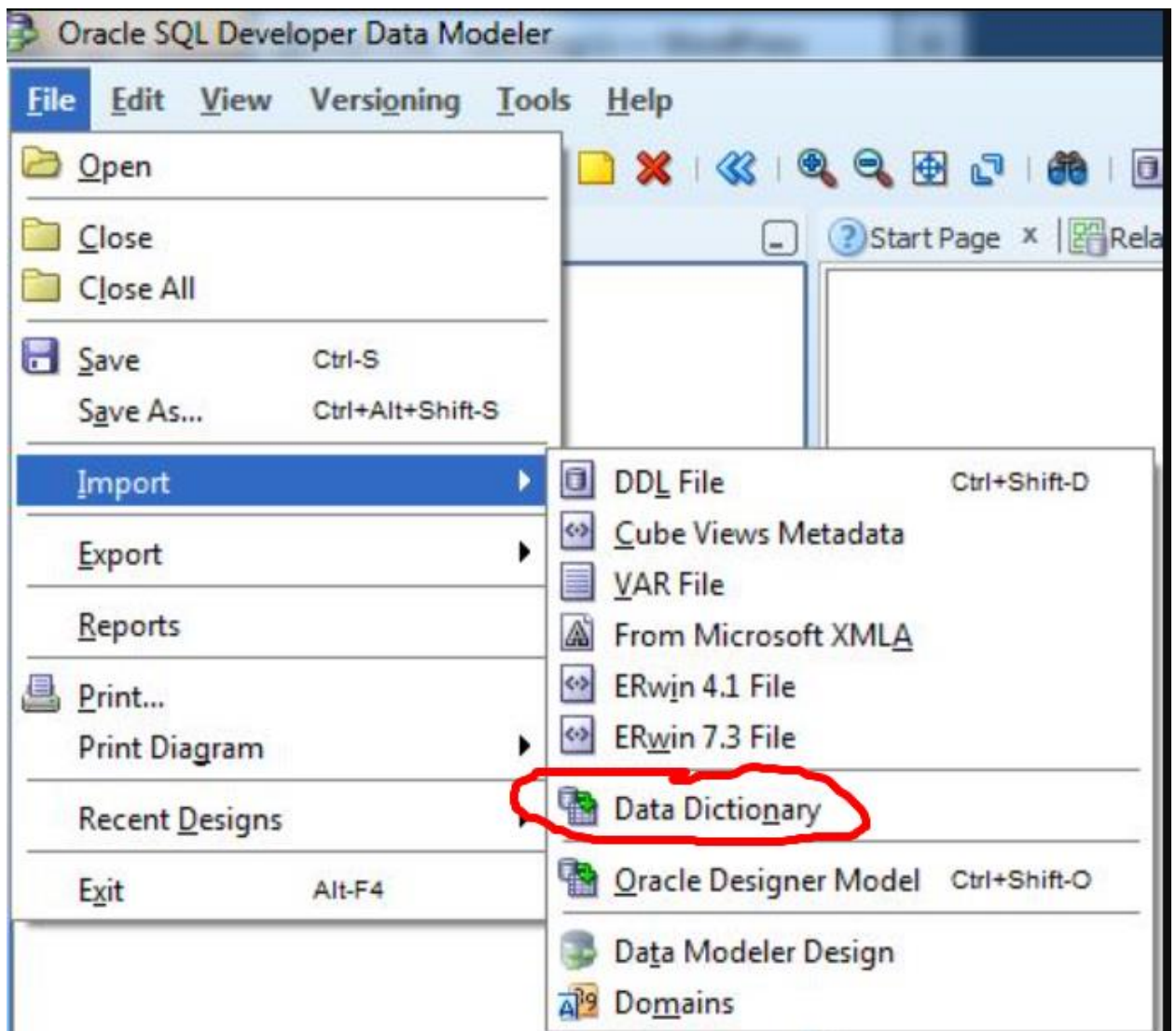


Рисунок 2.8 – Імпортування словника даних в Oracle SQL Developer Data Modeler

Після вибору бази даних, до якої необхідно здійснити під'єднання та введення всіх необхідних параметрів підключення (аналогічно рис. 2.3 з попереднього розділу) з'явиться можливість вибору схеми БД, базуючись на даних якої й буде необхідно виконати генерацію.

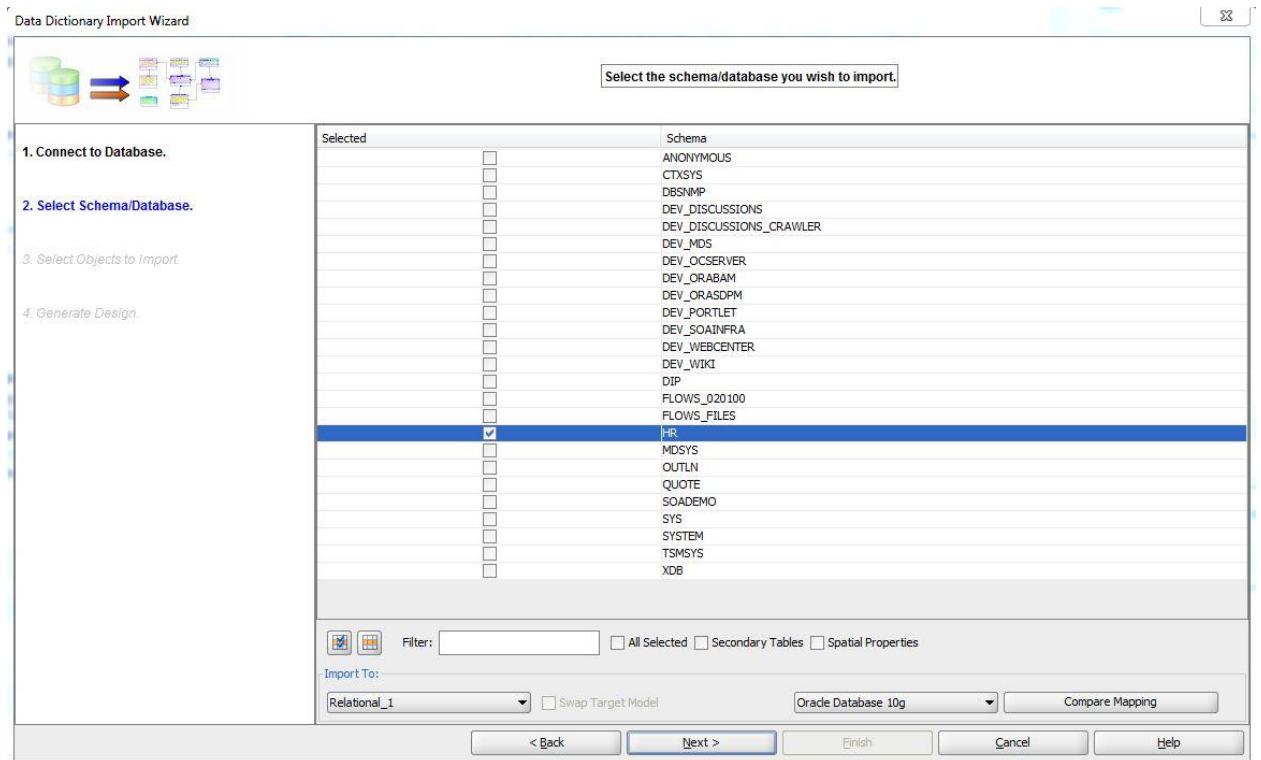


Рисунок 2.9 – Вибір схеми для імпортування в Oracle SQL Developer Data Modeler

І також необхідно, на наступному кроці обрати таблиці, для яких і буде згенерована ERD. Data Modeler має певну гнучкість і дає можливість довільно обирати таблиці зі схеми, не генеруючи діаграму відразу для всіх таблиць.

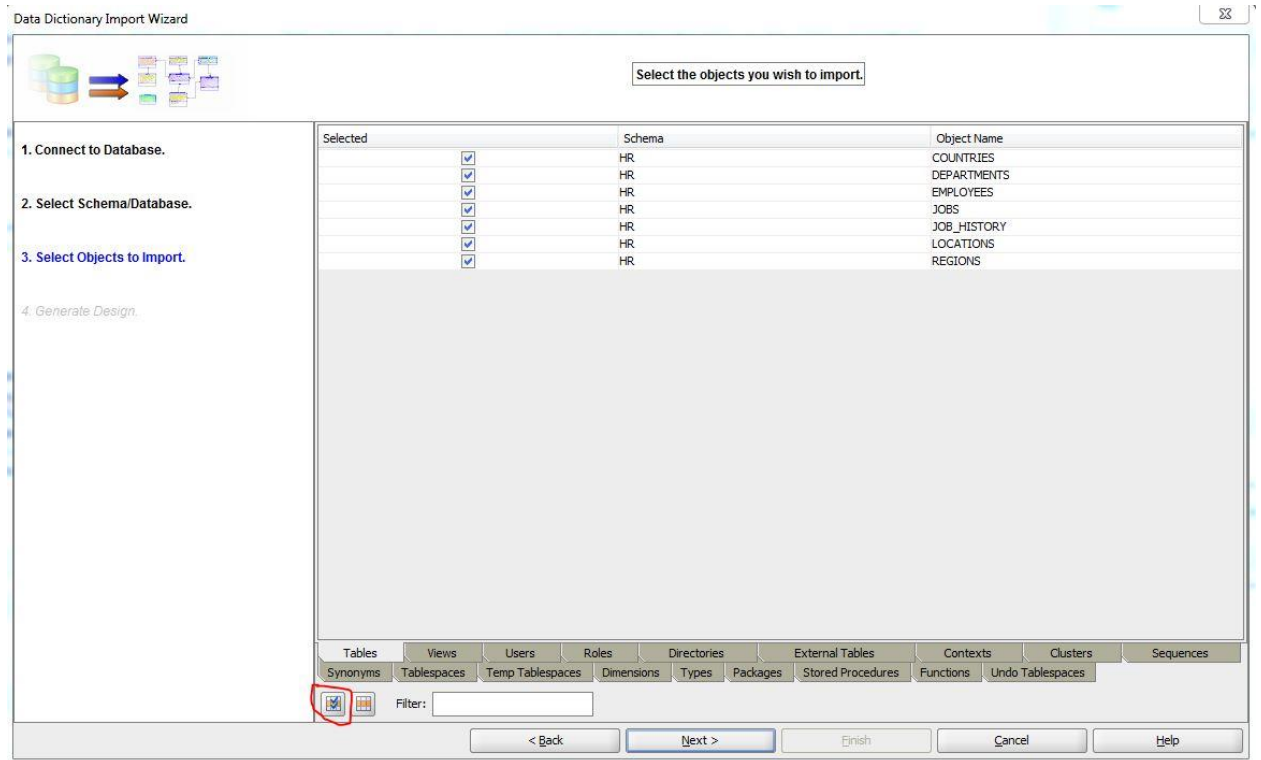


Рисунок 2.10 – Вибір необхідних таблиць зі схеми БД для побудови діаграми в Oracle SQL Developer Data Modeler

В результаті успішної генерації ERD Data Modeler дає змогу чітко побачити відношення між таблицями та їх структуру, що і зображено на рис. 2.11.

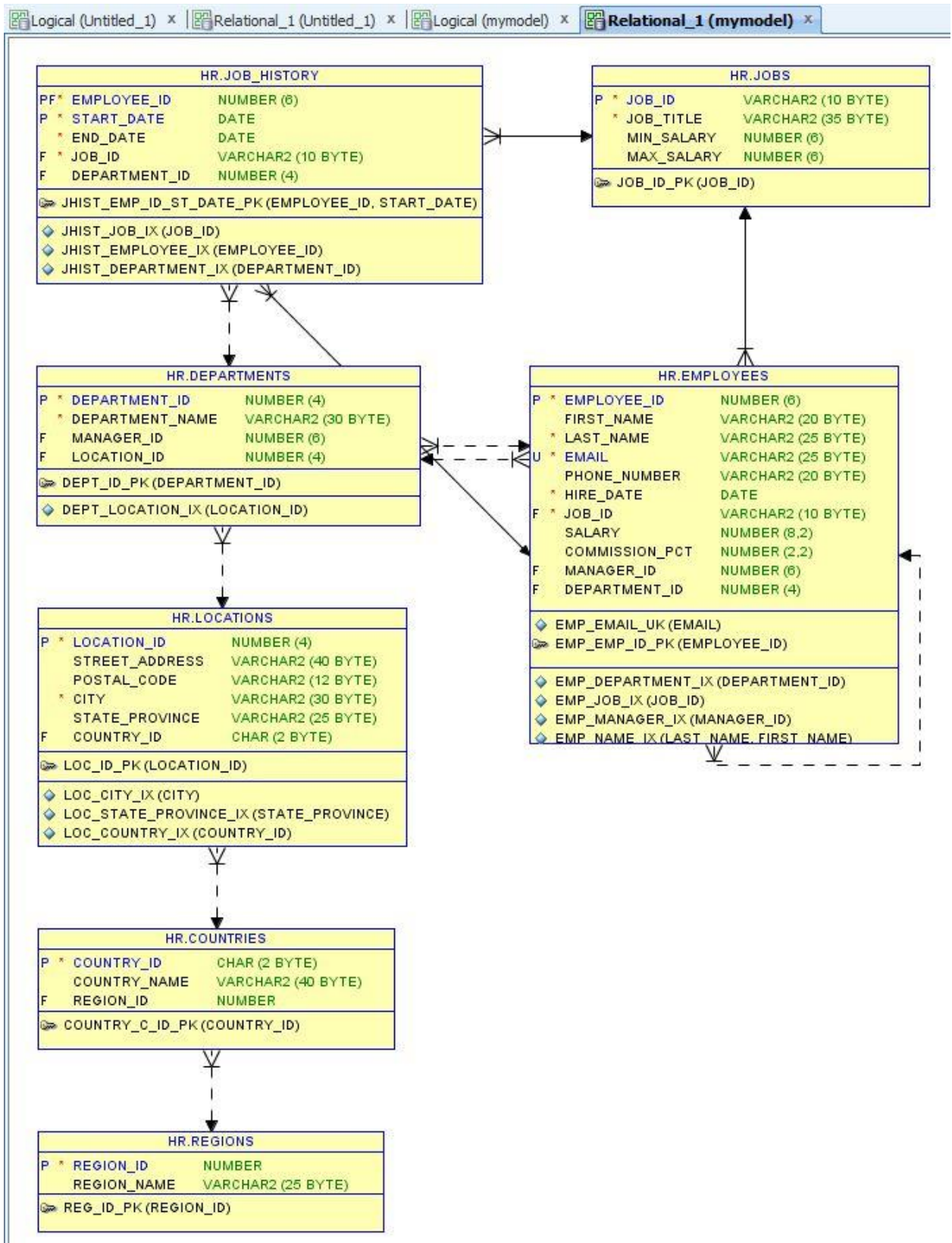


Рисунок 2.11 – Автоматично згенерована ERD в середовищі Oracle SQL Developer Data Modeler

Маючи в руках такий достатньо потужний інструмент як Oracle SQL Developer Data Modeler процес проектування та розробки нової БД стає значно простішим. Також зменшуються витрати на ознайомлення та аналіз структур уже наявної схеми БД. Це дає змогу в разі підвищити ефективність роботи користувача. Однак, опанування такими потужними інструментами як Oracle SQL Developer і Oracle SQL Developer Data Modeler потребують певного часу, але це не від'ємна частина при набутті нових навичок та умінь.

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 3.1 Візуалізація схеми бази даних

Згенеровано автоматично відносини і їх зв'язки за допомогою інструмента SQL Developer Data Modeler для досліджуваної бази даних (див. рис. 3.1).

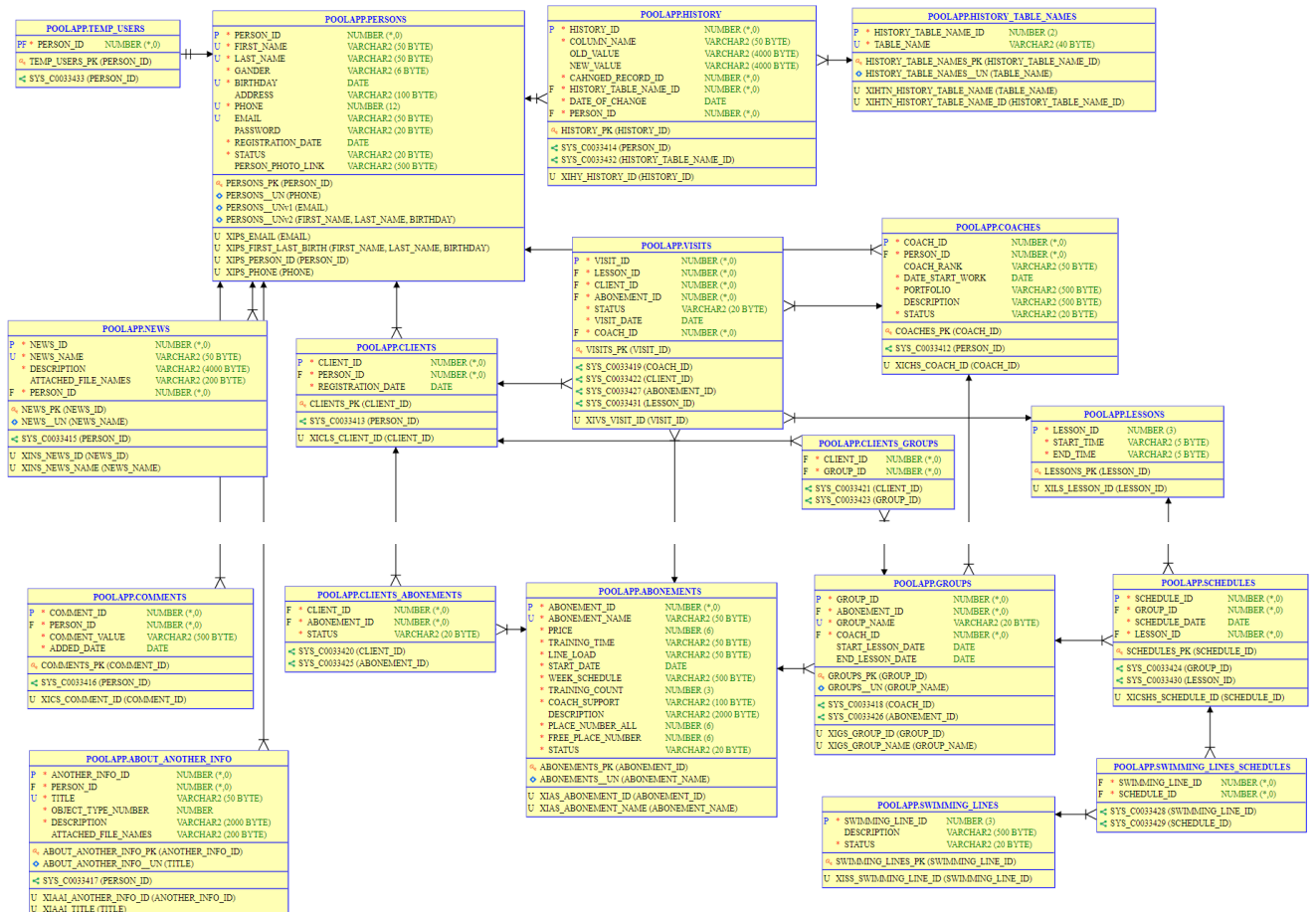


Рисунок 3.1 - ERD Інформаційної системи обліку користувачів басейну

### 3.2 Базова оптимізація запитів

В рамках базової оптимізації можна виділити декілька основних частин:

- Виокремлення найдовших запитів за часом;
- Створення додаткових необхідних індексів;
- Фіксація статистики Oracle.

Буде доцільно показати на прикладі одного з запитів динаміку його поведінки починаючи з плану запиту без додаткових індексів. Для тестового приклада взято запит котрий виводить активні абонементи для користувача, групи та тренерів пов'язані з цими абонементом:

```
select c.person_id, a.abonement_id, a.abonement_name, a.start_date,  
a.week_schedule, a.status, g.group_id, g.group_name, cs.coach_id,  
cp.first_name, cp.last_name  
from poolapp.clients c, poolapp.clients_abonements ca,  
poolapp.abonements a, poolapp.groups g, poolapp.coaches cs,  
poolapp.persons cp  
where c.person_id = 91000  
and c.client_id = ca.client_id  
and ca.abonement_id = a.abonement_id  
and ca.status = 'ACTIVE'  
and a.abonement_id = g.abonement_id  
and cs.coach_id = g.coach_id(+)  
and cp.person_id = cs.person_id(+)
```



Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	1580	247 (2)	00:00:01
1	NESTED LOOPS		10	1580	247 (2)	00:00:01
2	NESTED LOOPS		10	1580	247 (2)	00:00:01
* 3	HASH JOIN		10	1320	237 (2)	00:00:01
* 4	HASH JOIN		10	1230	230 (2)	00:00:01
5	NESTED LOOPS		1	91	207 (1)	00:00:01

PLAN\_TABLE\_OUTPUT

6	NESTED LOOPS		1	91	207 (1)	00:00:01
* 7	HASH JOIN		1	30	206 (1)	00:00:01
* 8	TABLE ACCESS FULL	CLIENTS	1	10	103 (1)	00:00:01
* 9	TABLE ACCESS FULL	CLIENTS_ABONEMENTS	1658	33160	103 (1)	00:00:01
* 10	INDEX UNIQUE SCAN	XIAS_ABONEMENT_ID	1		0 (0)	00:00:01
11	TABLE ACCESS BY INDEX ROWID	ABONEMENTS	1	61	1 (0)	00:00:01
12	TABLE ACCESS FULL	GROUPS	10000	312K	22 (0)	00:00:01
13	TABLE ACCESS FULL	COACHES	998	8982	7 (0)	00:00:01
* 14	INDEX UNIQUE SCAN	XIPS_PERSON_ID	1		0 (0)	00:00:01
15	TABLE ACCESS BY INDEX ROWID	PERSONS	1	26	1 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

Predicate Information (identified by operation id):

```

3 - access("CS"."COACH_ID"="G"."COACH_ID")
4 - access("A"."ABONEMENT_ID"="G"."ABONEMENT_ID")
7 - access("C"."CLIENT_ID"="CA"."CLIENT_ID")
8 - filter("C"."PERSON_ID"=91000)
9 - filter("CA"."STATUS"='ACTIVE')
10 - access("CA"."ABONEMENT_ID"="A"."ABONEMENT_ID")
14 - access("CP"."PERSON_ID"="CS"."PERSON_ID")

```

PLAN\_TABLE\_OUTPUT

37 rows selected.

Рисунок 3.2 – План запиту для отримання інформації про активні абонементи користувача

Аналізуючи план запиту зображеного на рис 3.2 можна легко помітити чисельні неефективні операції з повним зчитуванням таблиць (TABLE ACCESS FULL), а отже створення індексів для можливої оптимізації неминучий крок. Зокрема можна виділити необхідність створення наступних індексів:

- Індекс XICLS\_PERSON\_ID для колонки PERSON\_ID в таблиці CLIENTS;
- Індекс XIGS\_ABONEMENT\_ID для колонки ABONEMENT\_ID в таблиці GROUPS.

Створення індексів XICLS\_PERSON\_ID та XIGS\_ABONEMENT\_ID виконано наступним чином:

```

SQL>
SQL> CREATE INDEX POOLAPP.XICLS_PERSON_ID ON POOLAPP.CLIENTS (PERSON_ID);
■
Index created.

SQL> CREATE INDEX POOLAPP.XIGS_ABONEMENT_ID ON POOLAPP.GROUPS (ABONEMENT_ID);
Index created.

SQL>

```

Рисунок 3.3 - Створення індексів XICLS\_PERSON\_ID та XIGS\_ABONEMENT\_ID

Повний скрипт для створення всіх додаткових індексів наведено в додатку 1.

Для фіксації постійного плану запиту, незалежного від зміни кількості даних в таблицях та індексах згенерована штучна таблиця зі статистикою, котра базується на максимальних показниках закладених для поточної бази даних. Скрипт створення та застосування нової, синтетично згенерованої статистики наведено в додатку 2. Автоматичний збір статистики ORACLE також вимкнено.

```

SQL>
SQL> begin
  2  dbms_auto_task_admin.disable(
  3  client_name=>'auto optimizer stats collection',
  4  operation=>null,
  5  window_name=>null);
  6  end;
  7  /

PL/SQL procedure successfully completed.

```

Рисунок 3.4 – Вимкнення автоматичного збору статистики ORACLE

Виконавши створення додаткових індексів та фіксації статистики початковий тестовий план запиту для отримання інформації про активні абонементи користувача має наступний вигляд:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	1580	127 (2)	00:00:01
1	NESTED LOOPS		10	1580	127 (2)	00:00:01
2	NESTED LOOPS		10	1580	127 (2)	00:00:01
* 3	HASH JOIN		10	1320	117 (2)	00:00:01
4	NESTED LOOPS		10	1230	110 (2)	00:00:01
5	NESTED LOOPS		10	1230	110 (2)	00:00:01

PLAN\_TABLE\_OUTPUT

6	NESTED LOOPS		1	91	107 (2)	00:00:01
* 7	HASH JOIN		1	30	106 (2)	00:00:01
8	TABLE ACCESS BY INDEX ROWID BATCHED	CLIENTS	1	10	2 (0)	00:00:01
* 9	INDEX RANGE SCAN	XICLS_PERSON_ID	1		1 (0)	00:00:01
* 10	TABLE ACCESS FULL	CLIENTS_ABONEMENTS	50000	976K	103 (1)	00:00:01
11	TABLE ACCESS BY INDEX ROWID	ABONEMENTS	1	61	1 (0)	00:00:01
* 12	INDEX UNIQUE SCAN	XIAS_ABONEMENT_ID	1		0 (0)	00:00:01
* 13	INDEX RANGE SCAN	XIGS_ABONEMENT_ID	10		1 (0)	00:00:01
14	TABLE ACCESS BY INDEX ROWID	GROUPS	10	320	3 (0)	00:00:01
15	TABLE ACCESS FULL	COACHES	1000	9000	7 (0)	00:00:01
* 16	INDEX UNIQUE SCAN	XIPS_PERSON_ID	1		0 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

17	TABLE ACCESS BY INDEX ROWID	PERSONS	1	26	1 (0)	00:00:01
----	-----------------------------	---------	---	----	-------	----------

Predicate Information (identified by operation id):

```

3 - access("CS"."COACH_ID"="G"."COACH_ID")
7 - access("C"."CLIENT_ID"="CA"."CLIENT_ID")
9 - access("C"."PERSON_ID"=91000)
10 - filter("CA"."STATUS"='ACTIVE')
12 - access("CA"."ABONEMENT_ID"="A"."ABONEMENT_ID")

```

PLAN\_TABLE\_OUTPUT

```

13 - access("A"."ABONEMENT_ID"="G"."ABONEMENT_ID")
16 - access("CP"."PERSON_ID"="CS"."PERSON_ID")

```

Note

39 rows selected.

Рисунок 3.5 – План запиту для отримання інформації про активні абонементи користувача після створення індексів та фіксації статистики

Аналізуючи план запиту зображеного на рис 3.5 після створення індексів та фіксації статистики, легко помітити значне покращення плану запиту, зменшення числа повних зчитування таблиць завдяки використанню індексів. Попри те ще досі можна помітити повне зчитування таблиці COACHES і не використання жодного з індексів для доступу до неї, а отже потрібно розглянути подальші кроки для оптимізації описані в наступному розділі.

### 3.3 Поглиблена оптимізація запитів

Для досягнення кращої продуктивності запитів, в рамках поглибленої оптимізації використовувались підказки оптимізатору для примусового використання наявних індексів, завантаження таблиць в певному порядку та з певним типом з'єднання між ними. Також використана підказка GATHER\_PLAN\_STATISTICS для збирання додаткових показників під час

виконання запити. Для прикладу такої оптимізації взято запит, котрий уже розглядався в базові оптимізації, а також інший запит який також потребує оптимізації. Зокрема на рис 3.6 зображено план виконання запити для отримання інформації про активні абонементи користувача з використанням підказки GATHER\_PLAN\_STATISTICS, що дає змогу отримати фактичну кількість рядків для кожної ітерації запити.

```

| Id | Operation | Name | Starts | E-Rows | A-Rows | A-Time | Buffers | OMem | lMem | Used-Mem |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | SELECT STATEMENT | | 1 | | 10 | 00:00:00.01 | 426 | | | |
| 1 | NESTED LOOPS | | 1 | 10 | 10 | 00:00:00.01 | 426 | | | |
| 2 | NESTED LOOPS | | 1 | 10 | 10 | 00:00:00.01 | 416 | | | |
|* 3 | HASH JOIN | | 1 | 10 | 10 | 00:00:00.01 | 407 | 884K | 884K | 467K (0) |
| 4 | NESTED LOOPS | | 1 | 10 | 10 | 00:00:00.01 | 383 | | | |

PLAN_TABLE_OUTPUT
| 5 | NESTED LOOPS | | 1 | 10 | 10 | 00:00:00.01 | 382 | | | |
| 6 | NESTED LOOPS | | 1 | 1 | 1 | 00:00:00.01 | 380 | | | |
|* 7 | HASH JOIN | | 1 | 1 | 1 | 00:00:00.01 | 377 | 1856K | 1856K | 761K (0) |
| 8 | TABLE ACCESS BY INDEX ROWID BATCHED | CLIENTS | 1 | 1 | 1 | 00:00:00.01 | 3 | | | |
|* 9 | INDEX RANGE SCAN | XICLS_PERSON_ID | 1 | 1 | 1 | 00:00:00.01 | 2 | | | |
|* 10 | TABLE ACCESS FULL | CLIENTS_ABONEMENTS | 1 | 50000 | 1658 | 00:00:00.01 | 374 | | | |
| 11 | TABLE ACCESS BY INDEX ROWID | ABONEMENTS | 1 | 1 | 1 | 00:00:00.01 | 3 | | | |
|* 12 | INDEX UNIQUE SCAN | XIAS_ABONEMENT_ID | 1 | 1 | 1 | 00:00:00.01 | 2 | | | |
|* 13 | INDEX RANGE SCAN | XIGS_ABONEMENT_ID | 1 | 10 | 10 | 00:00:00.01 | 2 | | | |
| 14 | TABLE ACCESS BY INDEX ROWID | GROUPS | 10 | 10 | 10 | 00:00:00.01 | 1 | | | |
| 15 | TABLE ACCESS FULL | COACHES | 1 | 1000 | 998 | 00:00:00.01 | 24 | | | |

PLAN_TABLE_OUTPUT
|* 16 | INDEX UNIQUE SCAN | XIPS_PERSON_ID | 10 | 1 | 10 | 00:00:00.01 | 9 | | | |
| 17 | TABLE ACCESS BY INDEX ROWID | PERSONS | 10 | 1 | 10 | 00:00:00.01 | 10 | | | |

Predicate Information (identified by operation id):
3 - access("CS"."COACH_ID"="G"."COACH_ID")
7 - access("C"."CLIENT_ID"="CA"."CLIENT_ID")
9 - access("C"."PERSON_ID"=91000)
10 - filter("CA"."STATUS"='ACTIVE')

PLAN_TABLE_OUTPUT
12 - access("CA"."ABONEMENT_ID"="A"."ABONEMENT_ID")
13 - access("A"."ABONEMENT_ID"="G"."ABONEMENT_ID")
16 - access("CP"."PERSON_ID"="CS"."PERSON_ID")

52 rows selected.

```

Рисунок 3.6 – Розширений план запити для отримання інформації про активні абонементи користувача

Як було встановлено раніше, проблема запити в повному зчитуванні таблиці COACHES, хоча як можна переконатись на рис. 3.6 для предиката “3 - access("CS"."COACH\_ID"="G"."COACH\_ID")” фактична кількість рядків після виконання операції HASH JOIN – 10 при розмірі таблиці COACHES 1000 рядків, що становить лише 1% від загальної кількості, а отже доцільно використати індекс XICHS\_COACH\_ID. Тоді досліджуваний запит з підказкою для використання індексу XICHS\_COACH\_ID матиме наступний вигляд:

```

select /*+gather_plan_statistics index(cs XICHS_COACH_ID) */
c.person_id, a.abonement_id,a.abonement_name,
a.start_date, a.week_schedule, a.status, g.group_id, g.group_name,
cs.coach_id, cp.first_name, cp.last_name
from poolapp.clients c, poolapp.clients_abonements ca,
poolapp.abonements a, poolapp.groups g,
poolapp.coaches cs, poolapp.persons cp
where c.person_id = 91000
and c.client_id = ca.client_id
and ca.abonement_id = a.abonement_id
and ca.status = 'ACTIVE'
and a.abonement_id = g.abonement_id
and cs.coach_id = g.coach_id(+)
and cp.person_id = cs.person_id(+)

```

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	OMem	lMem	Used-Mem
0	SELECT STATEMENT		1		10	00:00:00.01	421			
1	NESTED LOOPS		1	10	10	00:00:00.01	421			
2	NESTED LOOPS		1	10	10	00:00:00.01	411			
3	NESTED LOOPS		1	10	10	00:00:00.01	402			
4	NESTED LOOPS		1	10	10	00:00:00.01	383			
PLAN_TABLE_OUTPUT										
5	NESTED LOOPS		1	1	1	00:00:00.01	380			
* 6	HASH JOIN		1	1	1	00:00:00.01	377	1856K	1856K	777K (0)
7	TABLE ACCESS BY INDEX ROWID BATCHED	CLIENTS	1	1	1	00:00:00.01	3			
* 8	INDEX RANGE SCAN	XICLS_PERSON_ID	1	1	1	00:00:00.01	2			
* 9	TABLE ACCESS FULL	CLIENTS_ABONEMENTS	1	50000	1658	00:00:00.01	374			
10	TABLE ACCESS BY INDEX ROWID	ABONEMENTS	1	1	1	00:00:00.01	3			
* 11	INDEX UNIQUE SCAN	XIAS_ABONEMENT_ID	1	1	1	00:00:00.01	2			
12	TABLE ACCESS BY INDEX ROWID BATCHED	GROUPS	1	10	10	00:00:00.01	3			
* 13	INDEX RANGE SCAN	XIGS_ABONEMENT_ID	1	10	10	00:00:00.01	2			
14	TABLE ACCESS BY INDEX ROWID	COACHES	10	1	10	00:00:00.01	19			
* 15	INDEX UNIQUE SCAN	XICHS_COACH_ID	10	1	10	00:00:00.01	9			
PLAN_TABLE_OUTPUT										
* 16	INDEX UNIQUE SCAN	XIPS_PERSON_ID	10	1	10	00:00:00.01	9			
17	TABLE ACCESS BY INDEX ROWID	PERSONS	10	1	10	00:00:00.01	10			
Predicate Information (identified by operation id):										
6 - access("C"."CLIENT_ID"="CA"."CLIENT_ID")										
8 - access("C"."PERSON_ID"=91000)										
9 - filter("CA"."STATUS"='ACTIVE')										
11 - access("CA"."ABONEMENT_ID"="A"."ABONEMENT_ID")										
PLAN_TABLE_OUTPUT										
13 - access("A"."ABONEMENT_ID"="G"."ABONEMENT_ID")										
15 - access("CS"."COACH_ID"="G"."COACH_ID")										
16 - access("CP"."PERSON_ID"="CS"."PERSON_ID")										
52 rows selected.										

Рисунок 3.7 – Розширений план запиту для отримання інформації про активні абонементи користувача після поглибленої оптимізації

За результатами виконання запиту зображеного на рис 3.7 показано, що використання індексу XICHS\_COACH\_ID дало змогу зменшити кількість зчитувань рядків таблиці до 10 без негативного впливу на інші операції, а отже

таким чином було досягнуто оптимального запиту для отримання інформації про активні абонементи користувача.

Аналізуючи проблемні запити, знайдені в попередніх кроках і виконуючи їх оптимізацію, застосування підказки з індексом це не єдиний випадок для покращення продуктивності. В ході аналізу знайдено складніший проблемний запит котрий відповідає за надання інформації про візити користувача, абонемент, групу та тренера котрий проводив заняття:

```
select  c.client_id,  a.abonement_id,  a.abonement_name,  g.group_id,
g.group_name, cs.coach_id, p.first_name, p.last_name,
v.visit_date, l.start_time, l.end_time
from  poolapp.clients  c,  poolapp.visits  v,  poolapp.abonements  a,
poolapp.groups  g,  poolapp.clients_groups  cg,
poolapp.coaches  cs,  poolapp.persons  p,  poolapp.lessons  l
where  c.client_id = 74974
and  c.client_id = v.client_id
and  v.abonement_id = a.abonement_id
and  c.client_id = cg.client_id
and  cg.group_id = g.group_id
and  v.coach_id = cs.coach_id
and  cs.person_id = p.person_id
and  v.lesson_id = l.lesson_id
```

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	OMem	lMem	Used-Mem
0	SELECT STATEMENT		1		100	00:00:00.01	445			
1	NESTED LOOPS		1	102	100	00:00:00.01	445			
2	NESTED LOOPS		1	102	100	00:00:00.02	345			
* 3	HASH JOIN		1	102	100	00:00:00.01	322	896K	896K	579K (0)
PLAN_TABLE_OUTPUT										
* 4	HASH JOIN		1	102	100	00:00:00.01	297	1538K	1538K	1280K (0)
5	TABLE ACCESS FULL	LESSONS	1	7	7	00:00:00.01	7			
* 6	HASH JOIN		1	102	100	00:00:00.01	287	1014K	1014K	518K (0)
7	MERGE JOIN CARTESIAN		1	102	100	00:00:00.01	264			
8	NESTED LOOPS		1	1	1	00:00:00.01	251			
9	NESTED LOOPS		1	1	1	00:00:00.01	250			
* 10	TABLE ACCESS FULL	CLIENTS_GROUPS	1	1	1	00:00:00.01	248			
* 11	INDEX UNIQUE SCAN	XIGS_GROUP_ID	1	1	1	00:00:00.01	2			
12	TABLE ACCESS BY INDEX ROWID	GROUPS	1	1	1	00:00:00.01	1			
13	BUFFER SORT		1	102	100	00:00:00.01	13	6144	6144	6144 (0)
* 14	TABLE ACCESS BY INDEX ROWID BATCHED	VISITS	1	102	100	00:00:00.01	13			
PLAN_TABLE_OUTPUT										
* 15	INDEX RANGE SCAN	XIVS_CLIENT_ID	1	102	100	00:00:00.01	3			
16	TABLE ACCESS FULL	ABONEMENTS	1	1000	1000	00:00:00.01	23			
17	TABLE ACCESS FULL	COACHES	1	1000	998	00:00:00.01	25			
* 18	INDEX UNIQUE SCAN	XIPS_PERSON_ID	100	1	1	00:00:00.01	23			
19	TABLE ACCESS BY INDEX ROWID	PERSONS	100	1	100	00:00:00.01	100			
Predicate Information (identified by operation id):										
3	- access("V"."COACH_ID"="CS"."COACH_ID")									
4	- access("V"."LESSON_ID"="L"."LESSON_ID")									
6	- access("V"."ABONEMENT_ID"="A"."ABONEMENT_ID")									
10	- filter("CG"."CLIENT_ID"=74974)									
11	- access("CG"."GROUP_ID"="G"."GROUP_ID")									
14	- filter("V"."LESSON_ID">=0)									
15	- access("V"."CLIENT_ID"=74974)									
18	- access("CS"."PERSON_ID"="P"."PERSON_ID")									
56 rows selected.										

Рисунок 3.8 – Розширений план запиту для отримання інформації про візити користувача

Аналізуючи розширений план запиту представлений на рис. 3.8 можна виокремити декілька проблем:

- Повне зчитування таблиці ABONEMENTS;
- Повне зчитування таблиці COACHES;
- Декартове з'єднання між результатами з таблиці VISITS та результатами з'єднань таблиць CLIENTS\_GROUPS та GROUPS ;
- Зайве з'єднання з таблицею CLIENTS.

Також під час тестових виконань запити для отримання інформації про візити користувача було знайдено не стабільну черговість читання даних, що може призвести до не стабільного плану виконання запита, а отже також потрібно чітко зафіксувати порядок виконання читання таблиць та індексів в цьому запиті. Після виконання всіх згаданих оптимізацій запит для отримання інформації про візити користувача має наступний вигляд:

```
select /*+ gather_plan_statistics leading(v,cg,a,g) use_nl(a,v) index(cs
XICHS_COACH_ID) index(g XIGS_GROUP_ID)*/
```

```

v.client_id,      a.abonement_id,      a.abonement_name,      g.group_id,
g.group_name, cs.coach_id, p.first_name, p.last_name,
v.visit_date, l.start_time, l.end_time
from poolapp.visits v, poolapp.abonements a, poolapp.groups g,
poolapp.clients_groups cg,
poolapp.coaches cs, poolapp.persons p, poolapp.lessons l
where v.client_id = 74974
and v.abonement_id = a.abonement_id
and v.client_id = cg.client_id
and cg.group_id = g.group_id
and v.coach_id = cs.coach_id
and cs.person_id = p.person_id
and v.lesson_id = l.lesson_id

```

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	OMem	IMem	Used-Mem
0	SELECT STATEMENT		1		100	00:00:00.03	621			
1	NESTED LOOPS		1	102	100	00:00:00.03	621			
2	NESTED LOOPS		1	102	100	00:00:00.04	521			
* 3	HASH JOIN		1	102	100	00:00:00.03	498	908K	908K	576K (0)
PLAN_TABLE_OUTPUT										
* 4	HASH JOIN		1	102	100	00:00:00.03	474	1572K	1572K	1224K (0)
5	TABLE ACCESS FULL	LESSONS	1	7	7	00:00:00.01	7			
* 6	HASH JOIN		1	102	100	00:00:00.03	464	1036K	1036K	486K (0)
7	NESTED LOOPS		1	102	100	00:00:00.01	370			
8	NESTED LOOPS		1	102	100	00:00:00.01	270			
* 9	HASH JOIN		1	102	100	00:00:00.01	261	1298K	1298K	821K (0)
* 10	TABLE ACCESS BY INDEX ROWID BATCHED	VISITS	1	102	100	00:00:00.01	13			
* 11	INDEX RANGE SCAN	XIVS_CLIENT_ID	1	102	100	00:00:00.01	3			
* 12	TABLE ACCESS FULL	CLIENTS_GROUPS	1	1	1	00:00:00.01	248			
* 13	INDEX UNIQUE SCAN	XIAS_ABONEMENT_ID	100	1	100	00:00:00.01	9			
14	TABLE ACCESS BY INDEX ROWID	ABONEMENTS	100	1	100	00:00:00.01	100			
PLAN_TABLE_OUTPUT										
15	TABLE ACCESS BY INDEX ROWID BATCHED	GROUPS	1	10000	10000	00:00:00.02	94			
16	INDEX FULL SCAN	XIGS_GROUP_ID	1	10000	10000	00:00:00.01	22			
17	TABLE ACCESS BY INDEX ROWID BATCHED	COACHES	1	1000	998	00:00:00.01	24			
18	INDEX FULL SCAN	XICHS_COACH_ID	1	1000	998	00:00:00.01	5			
* 19	INDEX UNIQUE SCAN	XIPS_PERSON_ID	100	1	100	00:00:00.01	23			
20	TABLE ACCESS BY INDEX ROWID	PERSONS	100	1	100	00:00:00.01	100			
Predicate Information (identified by operation id):										
3 - access("V"."COACH_ID"="CS"."COACH_ID")										
4 - access("V"."LESSON_ID"="L"."LESSON_ID")										
6 - access("CG"."GROUP_ID"="G"."GROUP_ID")										
9 - access("V"."CLIENT_ID"="CG"."CLIENT_ID")										
10 - filter("V"."LESSON_ID">=0)										
11 - access("V"."CLIENT_ID"=74974)										
12 - filter("CG"."CLIENT_ID"=74974)										
13 - access("V"."ABONEMENT_ID"="A"."ABONEMENT_ID")										
19 - access("CS"."PERSON_ID"="P"."PERSON_ID")										
58 rows selected.										

Рисунок 3.9 – Розширений план запиту для отримання інформації про візити користувача після оптимізації

Для підвищення продуктивності запиту зафіксовано чіткий порядок доступу до таблиць за допомогою підказки leading, а також окрім підказки для використання індексів XICHS\_COACH\_ID та XIGS\_GROUP\_ID застосовану підказку use\_nl котра дозволила змінити тип з'єднання між таблицями з HASH



JOIN на NESTED LOOPS, що виявилось ефективніше для запиту з для отримання інформації про візити користувача та отримати дані по індексу XIAS\_ABONEMENT\_ID, при унікальних даних в самому індексі. А отже використовуючи різні механізми оптимізації, не лише з підказками, а й з редагуванням самого вхідного запиту, вдалось отримати більш ефективний план запиту для отримання інформації про візити користувача, розширений план якого після оптимізації та зображено на рис. 3.9.

## ВИСНОВКИ

У кваліфікаційній магістерській роботі проведені дослідження підходів та методі підвищення ефективності роботи інформаційних систем. Обрано та проаналізовано головні переваги та недоліки використання цих методів.

Розроблені та впровадженні конкретні рекомендації підвищення ефективності роботи системи обліку користувачів басейну. Скорочено час відкліку, а отже і загалом покращена якість і зручність використання вебресурсу з обліку користувачів басейну.

Проаналізувавши роботу системи після виконання оптимізаційних заходів було знайдено декілька недоліків:

- Неефективне написання початкового запиту в додатку;
- Витрачання великої кількості часу на завантаження всього полотна запитів;
- Відсутність декількох необхідних колонок в таблицях.

Враховуючи вищесказане можна сформувати план подальшого покращення системи:

- Винесення початкових запитів з додатка в пакетні функції PL/SQL;
- Завантаження частини даних без очікування іншої;
- Додавання необхідних колонок до таблиць та покращення планів запитів шляхом використання нових доданих стовпців.

## СПИСОК ЛІТЕРАТУРИ

1. Коренев М.І. Інформаційна система обліку користувачів басейну товар [Текст]: робота на здобуття кваліфікаційного рівня бакалавр; спец.: 122 - комп'ютерні науки М.І Коренев; кер. Г.А. Олексієнко. - Суми: СумДУ, 2020
2. Cybersecurity Ventures [Електронний ресурс] – Режим доступу: <https://cybersecurityventures.com/our-company/>
3. Humans On The Internet Will Triple From 2015 To 2022 And Hit 6 Billion [Електронний ресурс] – Режим доступу: <https://cybersecurityventures.com/how-many-internet-users-will-the-world-have-in-2022-and-in-2030/>
4. Biswas R, Jiang Z, Kechezhi K, Knysh S, Mandra S, O’Gorman B, Perdomo-Ortiz A, Petukhov A, Realpe-Gomez J, Rieffel E, Venturelli D, Vasko F, Wang Z. A NASA perspective on quantum computing: Opportunities and challenges. Parallel Computing, 2017
5. Amagata D, Hara T, Nishio S. Sliding window top-k dominating query processing over distributed data streams. Distributed and Parallel Databases, 2016
6. Codd EF. Extending the Database Relation Model to Capture More Meaning. ACM Transaction on Database Systems., 1979
7. A Detailed Guide on SQL Query Optimization [Електронний ресурс] – Режим доступу: <https://www.analyticsvidhya.com/blog/2021/10/a-detailed-guide-on-sql-query-optimization/>
8. Niraj Gupta Oracle SQL and PL/SQL Performance Tuning: Query Tuning and Optimization - CreateSpace Independent Publishing Platform., 2015
9. Dan Tow - SQL Tuning: Generating Optimal Execution Plans: O'Reilly Media Inc, 2019

10. Query Optimizer Concepts [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/database/121/TGSQL/tgsql\\_optncpt.htm](https://docs.oracle.com/database/121/TGSQL/tgsql_optncpt.htm)
11. Optimizing SQL Statements [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/A81042\\_01/DOC/server.816/a76992/sql.htm](https://docs.oracle.com/cd/A81042_01/DOC/server.816/a76992/sql.htm)
12. SQL Tuning Overview [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/B19306\\_01/server.102/b14211/sql\\_1016.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14211/sql_1016.htm)
13. RAJEEV A. PARIDA THE ORACLE 11G PERFORMANCE TUNING., 2014
14. Jim Czuprynski, Deiby Gomez, Bert Scalzo Oracle Database 12c Release Testing Tools and Techniques for Performance and Scalability, 2017
15. Darl Kuhn, Sam Alapati, Bill Padfield Expert Indexing in Oracle Database 11g: Maximum Performance for your Database (Expert's Voice in Oracle), 2011
16. Using Optimizer Hints [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/B19306\\_01/server.102/b14211/hintsref.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14211/hintsref.htm)
17. Managing Optimizer Statistics [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/B19306\\_01/server.102/b14211/stats.htm#g49431](https://docs.oracle.com/cd/B19306_01/server.102/b14211/stats.htm#g49431)
18. Gathering Optimizer Statistics [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/A97630\\_01/server.920/a96533/stats.htm](https://docs.oracle.com/cd/A97630_01/server.920/a96533/stats.htm)
19. SQL\*Plus Overview [Электронный ресурс] – Режим доступа:  
[https://docs.oracle.com/cd/B14117\\_01/server.101/b12170/ch1.htm](https://docs.oracle.com/cd/B14117_01/server.101/b12170/ch1.htm)
20. Gennick Jonathan Oracle SQL\*Plus: The Definitive Guide, 2004
21. Getting Started with Oracle SQL Developer [Электронный ресурс] – Режим доступа:  
[https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/sqldev\\_mngdb/sqldev\\_mngdb\\_otn.htm](https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/sqldev_mngdb/sqldev_mngdb_otn.htm)
22. Heli Helskyaho Oracle SQL Developer Data Modeler for Database Design Mastery (Oracle Press), 2015

## Додаток 1. Створення додаткових індексів

```
-- PERSONS
CREATE INDEX POOLAPP.XIPS_STATUS ON POOLAPP.PERSONS (STATUS);
CREATE INDEX POOLAPP.XIPS_LAST_NAME ON POOLAPP.PERSONS (LAST_NAME);

-- NEWS
CREATE INDEX POOLAPP.XINS_PERSON_ID ON POOLAPP.NEWS (PERSON_ID);

-- COMMENTS
CREATE INDEX POOLAPP.XICS_PERSON_ID ON POOLAPP.COMMENTS (PERSON_ID);

-- ABOUT_ANOTHER_INFO
CREATE INDEX POOLAPP.XIAAI_PERSON_ID ON POOLAPP.ABOUT_ANOTHER_INFO
(PERSON_ID);
CREATE INDEX POOLAPP.XICLS_PERSON_ID ON POOLAPP.CLIENTS (PERSON_ID);

-- ABONEMENTS
CREATE INDEX POOLAPP.XIAS_STATUS ON POOLAPP.ABONEMENTS (STATUS);

-- COACHES
CREATE INDEX POOLAPP.XICHS_PERSON_ID ON POOLAPP.COACHES (PERSON_ID);
CREATE INDEX POOLAPP.XICHS_STATUS ON POOLAPP.COACHES (STATUS);

-- GROUPS
CREATE INDEX POOLAPP.XIGS_COACH_ID ON POOLAPP.GROUPS (COACH_ID);
CREATE INDEX POOLAPP.XIGS_ABONEMENT_ID ON POOLAPP.GROUPS
(ABONEMENT_ID);

-- SCHEDULES
CREATE INDEX POOLAPP.XICSHS_GROUP_ID ON POOLAPP.SCHEDULES (GROUP_ID);
CREATE INDEX POOLAPP.XICSHS_LESSON_ID ON POOLAPP.SCHEDULES
```

```
(LESSON_ID);
```

```
-- VISITS
```

```
CREATE INDEX POOLAPP.XIVS_LESSON_ID ON POOLAPP.VISITS (LESSON_ID);
```

```
CREATE INDEX POOLAPP.XIVS_CLIENT_ID ON POOLAPP.VISITS (CLIENT_ID);
```

```
CREATE INDEX POOLAPP.XIVS_ABONEMENT_ID ON POOLAPP.VISITS  
(ABONEMENT_ID);
```

```
CREATE INDEX POOLAPP.XIVS_COACH_ID ON POOLAPP.VISITS (COACH_ID);
```

```
-- HISTORY
```

```
CREATE INDEX POOLAPP.XIHY_PERSON_ID ON POOLAPP.HISTORY (PERSON_ID);
```

```
CREATE INDEX POOLAPP.XIHY_HISTORY_TABLE_NAME_ID ON POOLAPP.HISTORY  
(HISTORY_TABLE_NAME_ID);
```

## Додаток 2. Фіксація статистики Oracle

```

-- disable stats collection
begin
dbms_auto_task_admin.disable(
client_name=>'auto optimizer stats collection',
operation=>null,
window_name=>null);
end;
/
--drop and create STAT_TABLE
EXEC DBMS_STATS.DROP_STAT_TABLE('POOLAPP', 'POOL_STATS_TABLE');
EXEC DBMS_STATS.CREATE_STAT_TABLE('POOLAPP', 'POOL_STATS_TABLE');

REM INSERTING into poolapp.POOL_STATS_TABLE
SET DEFINE OFF;
-- ABONEMENTS
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','ABONEMENTS',null,null,null,'POOLAPP',null,'1000','2
0','139','1000',null,null,null,null,'0',null,null,null,null,to_date('0
8.05.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','5000','ABONEMENTS',null,null,'STATUS','POOLAPP',null,'3
','0,0005','3','1000','0','3598559201397120000000000000000000000000', '4169
303858776810000000000000000000000000','7','1','1000','41693038587768100000
000000000000000000','0',null,to_date('08.05.22','DD.MM.RR'),null,'454E44

```

```

4544', '504C414E4E4544', null, null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'ABONEMENTS', null, null, 'FREE_PLACE_NUMBER', 'POOLAPP',
, null, '6', '0,1666666666666667', '6', '1000', '0', '0', '40', '3', null, null, nu
ll, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '80', 'C129', null, null
, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'ABONEMENTS', null, null, 'PLACE_NUMBER_ALL', 'POOLAPP',
, null, '7', '0,142857142857143', '7', '1000', '0', '20', '90', '3', null, null, nu
ll, null, null, to_date('08.05.22', 'DD.MM.RR'), null, 'C115', 'C15B', null, nu
ll, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'ABONEMENTS', null, null, 'DESCRIPTION', 'POOLAPP', null,
'1000', '0,001', '1000', '1000', '0', '433220653891256000000000000000000000
', '433220653891256000000000000000000000000000000000000000000000000000000000
', '21', null, null, null, null, null
, to_date('08.05.22', 'DD.MM.RR'), null, '536F6D65206465736372697074696F6E
2030', '536F6D65206465736372697074696F6E20393939', null, null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'ABONEMENTS', null, null, 'COACH_SUPPORT', 'POOLAPP', nul
l, '981', '0,00101936799184506', '981', '1000', '0', '3651908298140390000000
00000000000000000000000000000000000000000000000000000000000000000000000000
', '453868702488952000000000000000000000000000000000000000000000000000000000
', '18', null, null,
null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '46554C4C2053555050
4F52542030', '576974686F757420535550504F5254', null, null, EMPTY_CLOB(),

```



```
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'TRAINING_COUNT','POOLAPP',nu
ll,'8','0,125','8','1000','0','10','50','3',null,null,null,null,null,t
o_date('08.05.22','DD.MM.RR'),null,'C10B','C133',null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'WEEK_SCHEDULE','POOLAPP',nul
l,'9','0,1111111111111111','9','1000','0','3651276871746350000000000000
00000000','4531330584421260000000000000000000000000','23',null,null,null,n
ull,null,to_date('08.05.22','DD.MM.RR'),null,'46522F53412F53414E203134
3A30302D31363A3030','57452F46522F53414E2031303A30302D31343A3030',null,
null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'START_DATE','POOLAPP',null,'
800','0,00125','800','1000','0','2457706','2459704','8',null,null,null
,null,null,to_date('08.05.22','DD.MM.RR'),null,'78740B0D010101','787A0
504010101',null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'LINE_LOAD','POOLAPP',null,'1
0','0,1','10','1000','0','255384724270128000000000000000000000000000000',
'29692
309913840700000000000000000000000000000000000000','20',null,null,null,null,to_date
('08.05.22','DD.MM.RR'),null,'312F706572736F6E7320666F72206C696E65','3
92F706572736F6E7320666F72206C696E65',null,null, EMPTY_CLOB(),
```

```

EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'TRAINING_TIME','POOLAPP',nul
l,'2','0,5','2','1000','0','26027215425778100000000000000000000000','270
656747974850000000000000000000000000','8',null,null,null,null,null,to_dat
e('08.05.22','DD.MM.RR'),null,'3220686F757273','3420686F757273',null,n
ull,EMPTY_CLOB(),EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'PRICE','POOLAPP',null,'965',
'0,00103626943005181','965','1000','0','104','10097','4',null,null,nul
l,null,null,to_date('08.05.22','DD.MM.RR'),null,'C20205','C3020162',nu
ll,null,EMPTY_CLOB(),EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABONEMENTS',null,null,'ABONEMENT_NAME','POOLAPP',nu
ll,'1000','0,001','1000','1000','0','33949580043793000000000000000000
000','339495800437930000000000000000000000','19',null,null,null,null,n
ull,to_date('08.05.22','DD.MM.RR'),null,'41626F6E656D656E74206E616D652
030','41626F6E656D656E74206E616D6520393939',null,null,EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','ABONEMENTS',null,null,'ABONEMENT_ID','POOLAPP',n
ull,'1000','0,001','1000','1000','0','1','1000','4',null,null,null,nul
l,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C20B',null,null,
EMPTY_CLOB(),EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE

```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIAS_ABONEMENT_ID',null,null,'ABONEMENTS','POOLAPP'
,'POOLAPP','1000','2','1000','1','1','21','1','1000',null,null,null,nu
ll,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'12.AB
ONEMENT_ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIAS_ABONEMENT_NAME',null,null,'ABONEMENTS','POOLAP
P','POOLAPP','1000','4','1000','1','1','951','1','1000',null,null,null
,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'14
.ABONEMENT_NAME', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIAS_STATUS',null,null,'ABONEMENTS','POOLAPP','POOL
APP','1000','3','3','1','20','60','1','1000',null,null,null,null,null,
to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'6.STATUS',
EMPTY_BLOB());
```

```
-- ABOUT_ANOTHER_INFO
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','ABOUT_ANOTHER_INFO',null,null,null,'POOLAPP',null,'
150000','1252','82','150000',null,null,null,null,'0',null,null,null,nu
ll,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,
EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','ABOUT_ANOTHER_INFO',null,null,'ATTACHED_FILE_NAMES'
```

```
, 'POOLAPP', null, '0', '0', '0', null, '150000', null, null, '0', null, null, null
, null, null, to_date('08.05.22', 'DD.MM.RR'), null, null, null, null, null,
EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
```

```
(null, 'C', '8', '2', 'ABOUT_ANOTHER_INFO', null, null, 'DESCRIPTION', 'POOLAP
P', null, '99584', '0,0000100417737789203', '99584', '150000', '0', '25508072
8899299000000000000000000000', '297121551100086400000000000000000000',
'38', null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '312
0736F6D65207465787420666F72206469736372697074696F6E203535343630', '3939
39393920736F6D65207465787420666F72206469736372697074696F6E203437333033
', null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
```

```
(null, 'C', '8', '2', 'ABOUT_ANOTHER_INFO', null, null, 'OBJECT_TYPE_NUMBER',
'POOLAPP', null, '100704', '0,00000993009215125516', '100704', '150000', '0'
, '1', '99999', '5', null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR
'), null, 'C102', 'C30A6464', null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
```

```
(null, 'C', '8', '2', 'ABOUT_ANOTHER_INFO', null, null, 'TITLE', 'POOLAPP', nul
l, '150000', '0,00000991316071216147', '150000', '150000', '0', '25508072889
9299000000000000000000000000', '297121551100086400000000000000000000', '29
', null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '312073
6F6D652072616E646F6D20696E666F203535343630', '393939393920736F6D6520726
16E646F6D20696E666F203437333033', null, null, EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
```

```
(null,'C','8','1000','ABOUT_ANOTHER_INFO',null,null,'PERSON_ID','POOLAPP',null,'100704','0,00000993009215125516','100704','150000','0','1','99999','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C30A6464',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','ABOUT_ANOTHER_INFO',null,null,'ANOTHER_INFO_ID','POOLAPP',null,'150000','0,00000991316071216147','150000','150000','0','1','101200','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C30B0D',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIAAI_ANOTHER_INFO_ID',null,null,'ABOUT_ANOTHER_INFO','POOLAPP','POOLAPP','150000','189','150000','1','1','1473','1','150000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,'15.ANOTHER_INFO_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIAAI_PERSON_ID',null,null,'ABOUT_ANOTHER_INFO','POOLAPP','POOLAPP','150000','224','100077','1','1','100026','1','150000',null,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,'9.PERSON_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIAAI_TITLE',null,null,'ABOUT_ANOTHER_INFO','POOLAPP','POOLAPP','150000','703','150000','1','1','100025','1','150000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null);
```

```

11,null,'5.TITLE', EMPTY_BLOB());

-- CLIENTS
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','CLIENTS',null,null,null,'POOLAPP',null,'100000','37
0','18','100000',null,null,null,null,'0',null,null,null,null,to_date('
08.05.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','CLIENTS',null,null,'REGISTRATION_DATE','POOLAPP',nu
ll,'1','1','1','100000','0','2459704,91865741','2459704,91865741','8',
null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'787A0504
170335','787A0504170335',null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','CLIENTS',null,null,'PERSON_ID','POOLAPP',null,'1
00000','0,0000100021004410926','100000','100000','0','1','100101','5',
null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C
30B0202',null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','CLIENTS',null,null,'CLIENT_ID','POOLAPP',null,'1
00000','0,0000100021004410926','100000','100000','0','1','100000','5',
null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C
30A6450',null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N

```

```

9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICLS_CLIENT_ID',null,null,'CLIENTS','POOLAPP','POO
LAPP','100000','187','100000','1','1','314','1','100000',null,null,nul
l,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'9
.CLIENT_ID', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICLS_PERSON_ID',null,null,'CLIENTS','POOLAPP','POO
LAPP','100000','222','100000','1','1','411','1','100000',null,null,nul
l,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'9
.PERSON_ID', EMPTY_BLOB());

-- CLIENTS_ABONEMENTS
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','CLIENTS_ABONEMENTS',null,null,null,'POOLAPP',null,'
100000','370','20','100000',null,null,null,null,'0',null,null,null,nul
l,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','100000','CLIENTS_ABONEMENTS',null,null,'STATUS','POOLAP
P',null,'2','0,00000511440933686569','2','100000','0','338864895110559
0000000000000000000000000000','406608130472475000000000000000000000','11','1'
,'10000','406608130472475000000000000000000000000000000000000000000000','0',null,to_date('08.0
5.22','DD.MM.RR'),null,'414354495645','4E4F5420414354495645','4E4F5420
414354495645',null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values

```

```
(null,'C','8','1000','CLIENTS_ABONEMENTS',null,null,'ABONEMENT_ID','POOLAPP',null,'759','0,00131752305665349','759','100000','0','1','1000','4',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C20B',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','CLIENTS_ABONEMENTS',null,null,'CLIENT_ID','POOLAPP',null,'100000','0,0000102288186737314','100000','100000','0','2294','100000','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C2175F','C30A6450',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
-- CLIENTS_GROUPS
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'T','8','2','CLIENTS_GROUPS',null,null,null,'POOLAPP',null,'100000','244','10','100000',null,null,null,null,'0',null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','CLIENTS_GROUPS',null,null,'GROUP_ID','POOLAPP',null,'745','0,00134228187919463','745','100000','0','40015','49995','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C305010','C3056460',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','CLIENTS_GROUPS',null,null,'CLIENT_ID','POOLAPP',null,'100000','0,0000104052858852297','100000','100000','0','2294','100000','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null
```



```
, 'C2175F', 'C30A6450', null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
-- COACHES
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null, 'T', '8', '2', 'COACHES', null, null, null, 'POOLAPP', null, '1000', '20', '113', '1000', null, null, null, null, '0', null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, null, null, null, null, EMPTY_CLOB(),
```

```
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null, 'C', '8', '5000', 'COACHES', null, null, 'STATUS', 'POOLAPP', null, '2', '0,000501002004008016', '2', '1000', '0', '40660813047247500000000000000000000000', '45333865697192000000000000000000000000', '11', '1', '1000', '45333865697192000000000000000000000000000000', '0', null, to_date('08.05.22', 'DD.MM.RR'),
```

```
null, '4E4F5420414354495645', '574F524B', null, null, EMPTY_CLOB(),
```

```
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null, 'C', '8', '2', 'COACHES', null, null, 'DESCRIPTION', 'POOLAPP', null, '1000', '0,00100200400801603', '1000', '1000', '0', '24988732563550100000000000000000000000', '29712154431534700000000000000000000000', '25', null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '30206578706972656E63656420636F6163682030', '393939206578706972656E63656420636F61636820393939', null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null, 'C', '8', '2', 'COACHES', null, null, 'PORTFOLIO', 'POOLAPP', null, '1000', '0,00100200400801603', '1000', '1000', '0', '24988589860010200000000000000000000000',
```

```
00000000', '297121544293572000000000000000000000', '35', null, null, null,
null, null, to_date('08.05.22', 'DD.MM.RR'), null, '30205375706572206368616
D70696F6E2E20546F7020706572736F6E2030', '393939205375706572206368616D70
696F6E2E20546F7020706572736F6E20393939', null, null, EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
(null, 'C', '8', '2', 'COACHES', null, null, 'DATE_START_WORK', 'POOLAPP', null
, '17', '0,0588235294117647', '17', '1000', '0', '2459704,95292824', '2459704
,95311343', '8', null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR')
, null, '787A050417350E', '787A050417351E', null, null, EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
(null, 'C', '8', '2', 'COACHES', null, null, 'COACH_RANK', 'POOLAPP', null, '100
0', '0,00100200400801603', '1000', '1000', '0', '43847316702214700000000000
000000000', '438473167022147000000000000000000000', '25', null, null, null
, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '5472656E65722068696768
2063617465676F72792030', '5472656E657220686967682063617465676F727920393
939', null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
(null, 'C', '8', '1000', 'COACHES', null, null, 'PERSON_ID', 'POOLAPP', null, '1
000', '0,00100200400801603', '1000', '1000', '0', '65', '99312', '5', null, nul
l, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, 'C142', 'C30A5E0D'
, null, null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID, TYPE, VERSION, FLAGS, C1, C2, C3, C4, C5, C6, N1, N2, N3, N4, N5, N6, N7, N8, N
9, N10, N11, N12, N13, D1, T1, R1, R2, R3, CH1, CL1, BL1) values
(null, 'C', '8', '1000', 'COACHES', null, null, 'COACH_ID', 'POOLAPP', null, '10
```

```
00','0,00100200400801603','1000','1000','0','1','1000','4',null,null,
null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C20B',null,
null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XICHS_COACH_ID',null,null,'COACHES','POOLAPP','POOL
APP','1000','2','1000','1','1','17','1','1000',null,null,null,null,nul
l,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'8.COACH_ID'
, EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XICHS_PERSON_ID',null,null,'COACHES','POOLAPP','POO
LAPP','1000','3','1000','1','1','946','1','1000',null,null,null,null,n
ull,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'9.PERSON_
ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XICHS_STATUS',null,null,'COACHES','POOLAPP','POOLAP
P','1000','4','2','2','10','20','1','1000',null,null,null,null,null,to
_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'6.STATUS',
EMPTY_BLOB());
```

```
--COMMENTS
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'T','8','2','COMMENTS',null,null,null,'POOLAPP',null,'100000','8
74','50','100000',null,null,null,null,'0',null,null,null,null,to_date(
'08.05.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
```

```
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','COMMENTS',null,null,'ADDED_DATE','POOLAPP',null,'1041','0,000960614793467819','1041','100000','0','2459704,91738426','2459704,9294213','8',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'787A0504170203','787A0504171317',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','COMMENTS',null,null,'COMMENT_VALUE','POOLAPP',null,'100000','0,0000100023005291217','100000','100000','0','2550807288992990000000000000000000000000','29712155110008640000000000000000000000','32',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'3120736F6D652072616E646F6D20636F6D6D656E742031','393939393920736F6D652072616E646F6D20636F6D6D656E74203939393939',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','COMMENTS',null,null,'PERSON_ID','POOLAPP',null,'100000','0,0000100023005291217','100000','100000','0','1','99999','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C30A6464',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','COMMENTS',null,null,'COMMENT_ID','POOLAPP',null,'100000','0,0000100023005291217','100000','100000','0','1','100100','5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C3
```

```

0B02',null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICS_COMMENT_ID',null,null,'COMMENTS','POOLAPP','PO
OLAPP','100000','187','100000','1','1','763','1','100000',null,null,nu
ll,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'
10.COMMENT_ID', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICS_PERSON_ID',null,null,'COMMENTS','POOLAPP','POO
LAPP','100000','222','100077','1','1','99743','1','100000',null,null,n
ull,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,
'9.PERSON_ID', EMPTY_BLOB());

-- GROUPS
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','GROUPS',null,null,null,'POOLAPP',null,'10000','73',
'48','10000',null,null,null,null,'0',null,null,null,null,to_date('08.0
5.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','6','GROUPS',null,null,'COACH_ID','POOLAPP',null,'1000',
'0,000837','1000','10000','0','5','1000','4','1','10000','1000','10',n
ull,to_date('08.05.22','DD.MM.RR'),null,'C106','C20B','C2092A',null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N

```

```

9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','GROUPS',null,null,'GROUP_NAME','POOLAPP',null,'1000
0','0,0001','10000','10000','0','25507724379109800000000000000000000000
','2971215442790510000000000000000000000000','19',null,null,null,null,null,
to_date('08.05.22','DD.MM.RR'),null,'312047726F7570206E616D65202B2031'
,'3939392047726F7570206E616D65202B2039',null,null,EMPTY_CLOB(),
EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','GROUPS',null,null,'ABONEMENT_ID','POOLAPP',null,
'1000','0,001','1000','10000','0','1','1000','4',null,null,null,null,n
ull,to_date('08.05.22','DD.MM.RR'),null,'C102','C20B',null,null,
EMPTY_CLOB(),EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','GROUPS',null,null,'GROUP_ID','POOLAPP',null,'100
00','0,0001','10000','10000','0','40015','50014','5',null,null,null,nu
ll,null,to_date('08.05.22','DD.MM.RR'),null,'C3050110','C306010F',null
,null,EMPTY_CLOB(),EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','GROUPS',null,null,'END_LESSON_DATE','POOLAPP',null,
'800','0,00125','800','10000','0','2457806','2459804','8',null,null,nu
ll,null,null,to_date('08.05.22','DD.MM.RR'),null,'78750215010101','787
A080C010101',null,null,EMPTY_CLOB(),EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','GROUPS',null,null,'START_LESSON_DATE','POOLAPP',nul
l,'800','0,00125','800','10000','0','2457706','2459704','8',null,null,

```

```
null,null,null,to_date('08.05.22','DD.MM.RR'),null,'78740B0D010101','787A0504010101',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIGS_ABONEMENT_ID',null,null,'GROUPS','POOLAPP','POOLAPP','10000','21','1000','1','1','1055','1','10000',null,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'12.ABONEMENT_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIGS_COACH_ID',null,null,'GROUPS','POOLAPP','POOLAPP','10000','21','749','1','1','861','1','10000',null,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'8.COACH_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIGS_GROUP_ID',null,null,'GROUPS','POOLAPP','POOLAPP','10000','19','10000','1','1','72','1','10000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'8.GROUP_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIGS_GROUP_NAME',null,null,'GROUPS','POOLAPP','POOLAPP','10000','50','10000','1','1','1168','1','10000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'10.GROUP_NAME',EMPTY_BLOB());
```

```
-- HISTORY
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','HISTORY',null,null,null,'POOLAPP',null,'100000000',
'455774','43','100000000',null,null,null,null,'0',null,null,null,null,
to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','4102','HISTORY',null,null,'PERSON_ID','POOLAPP',null,'1
','0,00000000733097144914299','1','5521','0','1','1','3','1','5521','1
','0',null,to_date('08.05.22','DD.MM.RR'),null,'C102','C102','C102',nu
ll,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','HISTORY',null,null,'DATE_OF_CHANGE','POOLAPP',null,
'23016','0,0000434480361487661','23016','100000000','0','2459704,87222
222','2459708,25084491','8',null,null,null,null,null,to_date('08.05.22
','DD.MM.RR'),null,'787A0504153901','787A050807020E',null,null,
EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','4102','HISTORY',null,null,'HISTORY_TABLE_NAME_ID','POOL
APP',null,'13','0,00000000735896243103971','13','100000000','0','1','1
5','3','1','100000000','15','0',null,to_date('08.05.22','DD.MM.RR'),nu
ll,'C102','C110','C110',null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','HISTORY',null,null,'CAHNGED_RECORD_ID','POOLAPP',nu
ll,'10818560','0,00000000924337434926645','10818560','100000000','0','1
```





```
959648','6',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,
'C102','C444606131',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,
N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIHY_HISTORY_ID',null,null,'HISTORY','POOLAPP','POOLAPP',
'100000000','151267','100000000','1','1','907739','2','505413',null,
null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,
null,null,'10.HISTORY_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,
N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIHY_HISTORY_TABLE_NAME_ID',null,null,'HISTORY','POOLAPP',
'POOLAPP','POOLAPP','100000000','132835','13','10218','34921','453976','2',
'68144317',null,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,
null,null,null,null,null,'21.HISTORY_TABLE_NAME_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,
N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIHY_PERSON_ID',null,null,'HISTORY','POOLAPP','POOLAPP',
'100000000','132835','1','132835','452623','452623','2','68144317',
null,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,
null,null,null,'9.PERSON_ID',EMPTY_BLOB());
```

```
-- HISTORY_TABLE_NAMES
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,
N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'T','8','2','HISTORY_TABLE_NAMES',null,null,null,'POOLAPP',null,
'15','5','23','15',null,null,null,null,'0',null,null,null,null,to_date(
'08.05.22','DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```



```

9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','LESSONS',null,null,null,'POOLAPP',null,'7','5','15',
,'7',null,null,null,null,'0',null,null,null,null,to_date('08.05.22','D
D.MM.RR'),null,null,null,null,null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','LESSONS',null,null,'END_TIME','POOLAPP',null,'7','0
,142857142857143','7','7','0','2553997611379660000000000000000000000000',
'255582302824399000000000000000000000000000000000000000000000000000',
'6',null,null,null,null,null,to_
date('08.05.22','DD.MM.RR'),null,'31302E3030','31392E3030',null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','LESSONS',null,null,'START_TIME','POOLAPP',null,'7',
'0,142857142857143','7','7','0','255399761137966000000000000000000000000000',
'291705670411287000000000000000000000000000000000000000000000000000',
'6',null,null,null,null,null,t
o_date('08.05.22','DD.MM.RR'),null,'31302E3030','382E3330',null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','LESSONS',null,null,'LESSON_ID','POOLAPP',null,'7
','0,142857142857143','7','7','0','1','7','3',null,null,null,null,null,
to_date('08.05.22','DD.MM.RR'),null,'C102','C108',null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XILS_LESSON_ID',null,null,'LESSONS','POOLAPP','POOL
APP','7','1','7','1','1','1','1','0','7',null,null,null,null,null,to_date(
'08.05.22','DD.MM.RR'),null,null,null,null,null,'9.LESSON_ID',

```

```
EMPTY_BLOB());
```

```
-- NEWS
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'T','8','2','NEWS',null,null,null,'POOLAPP',null,'5000','50','60',
'5000',null,null,null,null,'0',null,null,null,null,to_date('08.05.22',
'DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','5000','NEWS',null,null,'PERSON_ID','POOLAPP',null,'4','0,0001',
'4','5000','0','100','50000','3','1','5000','50000','0',null,to_date('08.05.22',
'DD.MM.RR'),null,'C202','C306','C306',null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','NEWS',null,null,'ATTACHED_FILE_NAMES','POOLAPP',null,'0','0','0',
null,'5000',null,null,'0',null,null,null,null,null,to_date('08.05.22',
'DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','NEWS',null,null,'DESCRIPTION','POOLAPP',null,'5000','0,0002',
'5000','5000','0','2498858967395640000000000000000000000000','2971215442935440000000000000000000000000',
'33',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'3020536F6D65206E657773206465736372697074696F6E732030',
'39393920536F6D65206E657773206465736372697074696F6E7320393939',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','NEWS',null,null,'NEWS_NAME','POOLAPP',null,'5000','0,0002','5000','5000','0','249885896739564000000000000000000000','2971215442935440000000000000000000000000','20',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'3020536F6D65206E6577732030','39393920536F6D65206E65777320393939',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','NEWS',null,null,'NEWS_ID','POOLAPP',null,'5000','0,0002','5000','5000','0','1','5000','4',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C233',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XINS_NEWS_ID',null,null,'NEWS','POOLAPP','POOLAPP','5000','9','5000','1','1','48','1','5000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'7.NEWS_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XINS_NEWS_NAME',null,null,'NEWS','POOLAPP','POOLAPP','5000','33','5000','1','1','978','1','5000',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'9.NEWS_NAME',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XINS_PERSON_ID',null,null,'NEWS','POOLAPP','POOLAPP','5000','10','4','2','45','180','1','5000',null,null,null,null,null,t
```

```
o_date('08.05.22', 'DD.MM.RR'), null, null, null, null, null, '9.PERSON_ID',  
EMPTY_BLOB());
```

```
-- PERSONS
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N  
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values  
(null,'T','8','2','PERSONS',null,null,null,'POOLAPP',null,'100000','17  
56','115','100000',null,null,null,null,'0',null,null,null,null,to_date  
( '08.05.22', 'DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),  
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N  
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values  
(null,'C','8','2','PERSONS',null,null,'PERSON_PHOTO_LINK','POOLAPP',nu  
ll,'100000','0,0000100021004410926','100000','100000','0','47997121508  
331200000000000000000000000000','479971215083312000000000000000000000','22'  
,null,null,null,null,null,to_date('08.05.22', 'DD.MM.RR'),null,'5C70686  
F746F735C313030303033235352E706E67','5C70686F746F735C3938383838333532  
362E706E67',null,null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N  
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values  
(null,'C','8','4102','PERSONS',null,null,'STATUS','POOLAPP',null,'3','  
0,00000500105022054631','3','100000','0','338884622913028000000000000000  
00000000','4430341650935480000000000000000000000000000','11','1','100000','44  
3034165093548000000000000000000000000000000000000000000000000000','0',null,to_date('08.05.22', 'DD.MM  
.RR'),null,'41444D494E','55534552','55534552',null, EMPTY_CLOB(),  
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N  
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values  
(null,'C','8','2','PERSONS',null,null,'REGISTRATION_DATE','POOLAPP',nu
```

```

ll, '1224', '0,000816993464052288', '1224', '100000', '0', '2459704,87061343
', '2459704,88636574', '8', null, null, null, null, null, to_date('08.05.22', '
DD.MM.RR'), null, '787A050415362A', '787A0504161117', null, null,
EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','PERSONS',null,null,'PASSWORD','POOLAPP',null,'49204
','0,0000203235509308186','49204','49567','50412','2502077023632490000
0000000000000000','635944284034538000000000000000000000000000000','9',null,nul
l,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'303031317A515873
4B54456334587A','7A7A79565149334A686F30316F3942',null,null,
EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','PERSONS',null,null,'EMAIL','POOLAPP',null,'49567','
0,0000201747130147074','49567','49567','50412','2502077023632490000000
0000000000000000','635944284034538000000000000000000000000000000','14',null,null,
null,null,null,to_date('08.05.22','DD.MM.RR'),null,'303031317A5158734B
54456334587A0676D61696C2E636F6D','7A7A79565149334A686F30316F394240676
D61696C2E636F6D',null,null, EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','PERSONS',null,null,'PHONE','POOLAPP',null,'100000',
'0,0000100021004410926','100000','100000','0','380100003255','38098888
3526','8',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null
,'C6270201012138','C6270A5959241B',null,null, EMPTY_CLOB(),
EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values

```



```
(null,'C','8','2','PERSONS',null,null,'ADDRESS','POOLAPP',null,'17630',
,'0,0000567214974475326','17630','25929','74050','43334235455261100000
0000000000000000','101000313942891000000000000000000000000000000000000',
'6',null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'53756D792C','C29
1D183D08CD0882C20C282D08ED181D08AD180D085D181D085C2ADD181D18CD08AC2A02
03130',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','PERSONS',null,null,'BIRTHDAY','POOLAPP',null,'26590',
,'0,0000376081233546446','26590','100000','0','2430330','2457703','8',
,null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'778D0C0
1010101','78740B0A010101',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','PERSONS',null,null,'GANDER','POOLAPP',null,'3','0,3
33333333333333','3','100000','0','36486638713733300000000000000000000000
','10103847580627000000000000000000000000000000000000000000000000000000',
'7',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'46454D414C45','C297D08ED08B',nul
l,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','6','PERSONS',null,null,'LAST_NAME','POOLAPP',null,'614',
,'0,001608','614','5549','0','3394757560331890000000000000000000000000000',
'108336087234153000000000000000000000000000000000000000000000000000000000',
'11','1','4885','448675773975110
000000000000000000000000000000000000000000000000000000000000000000000000',
'8',null,to_date('08.05.22','DD.MM.RR'),null,'4
161726F6E6F7669D0A4C28D','D0A5D09D6F7265736F7669D0A4C28D','56696C6F726
F7669D0A4C28D',null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','PERSONS',null,null,'FIRST_NAME','POOLAPP',null,'902',
'0,00110864745011086','902','10000','0','33949603410879400000000000000000',
'10833608723487800000000000000000000000000000000000000000000000000000',
'10',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'416272616D6F76','D0A5D09D757261766C657661',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','PERSONS',null,null,'PERSON_ID','POOLAPP',null,'100000',
'0,0000100021004410926','100000','100000','0','1','100101','5',
null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C30B0202',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIPS_EMAIL',null,null,'PERSONS','POOLAPP','POOLAPP',
'49567','313','49567','1','1','49531','1','49567',null,null,null,null,
null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'5.EMAIL',
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIPS_FIRST_LAST_BIRTH',null,null,'PERSONS','POOLAPP',
'POOLAPP','100000','1180','100000','1','1','99906','2','100000',null,
null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,
null,'10.FIRST_NAME,9.LAST_NAME,8.BIRTHDAY',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIPS_LAST_NAME',null,null,'PERSONS','POOLAPP','POOLAPP',
'100000','306','614','1','152','93752','1','100000',null,null,null,
null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'9
```

```

.LAST_NAME', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIPS_PERSON_ID',null,null,'PERSONS','POOLAPP','POOL
APP','100000','187','100000','1','1','1692','1','100000',null,null,nul
l,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'9
.PERSON_ID', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIPS_PHONE',null,null,'PERSONS','POOLAPP','POOLAPP'
,'100000','280','100000','1','1','99925','1','100000',null,null,null,n
ull,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'5.PH
ONE', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIPS_STATUS',null,null,'PERSONS','POOLAPP','POOLAPP
','100000','306','3','102','580','1740','1','100000',null,null,null,nu
ll,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'6.STA
TUS', EMPTY_BLOB());

-- SCHEDULES
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','SCHEDULES',null,null,null,'POOLAPP',null,'10000','5
0','21','10000',null,null,null,null,'0',null,null,null,null,to_date('0
8.05.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N

```

```

9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','4102','SCHEDULES',null,null,'LESSON_ID','POOLAPP',null,
'6','0,00005','6','10000','0','1','6','3','1','10000','6','0',null,to_
date('08.05.22','DD.MM.RR'),null,'C102','C107','C107',null,
EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','SCHEDULES',null,null,'SCHEDULE_DATE','POOLAPP',null
,'6','0,1666666666666667','6','10000','0','2459705,57728009','2459705,5
7734954','8',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),n
ull,'787A05050E3412','787A05050E3418',null,null, EMPTY_CLOB(),
EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','SCHEDULES',null,null,'GROUP_ID','POOLAPP',null,'
10000','0,0001','10000','10000','0','40015','50014','5',null,null,null
,null,null,to_date('08.05.22','DD.MM.RR'),null,'C3050110','C306010F',n
ull,null, EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','SCHEDULES',null,null,'SCHEDULE_ID','POOLAPP',nul
l,'10000','0,0001','10000','10000','0','14498','24497','5',null,null,n
ull,null,null,to_date('08.05.22','DD.MM.RR'),null,'C3022D63','C3032D62
',null,null, EMPTY_CLOB(), EMPTY_BLOB());

```

```

Insert into poolapp.POOL_STATS_TABLE

```

```

(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICSHS_GROUP_ID',null,null,'SCHEDULES','POOLAPP','P
OOLAPP','10000','23','10000','1','1','1052','1','10000',null,null,null
,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'8.

```

```

GROUP_ID', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICSHS_LESSON_ID',null,null,'SCHEDULES','POOLAPP','
POOLAPP','10000','20','6','3','7','45','1','10000',null,null,null,null
,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,null,'9.LESSO
N_ID', EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XICSHS_SCHEDULE_ID',null,null,'SCHEDULES','POOLAPP'
,'POOLAPP','10000','20','10000','1','1','42','1','10000',null,null,nul
l,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'1
1.SCHEDULE_ID', EMPTY_BLOB());

-- SWIMMING_LINES
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','SWIMMING_LINES',null,null,null,'POOLAPP',null,'10',
'5','15','10',null,null,null,null,'0',null,null,null,null,to_date('08.
05.22','DD.MM.RR'),null,null,null,null,null, EMPTY_CLOB(),
EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','SWIMMING_LINES',null,null,'STATUS','POOLAPP',null,'
2','0,5','2','10','0','3494316374475420000000000000000000000000000',
'4118195354755880000000000000000000000000000','6',null,null,null,null,
to_date('08.05.22','DD.MM.RR'),null,'434C4F534544','4F50454E',null,null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE

```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','2','SWIMMING_LINES',null,null,'DESCRIPTION','POOLAPP',null,'10','0','1','10','10','0','396752960651810000000000000000000000000000','3967529606518480000000000000000000000000000','8',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'4C696E652031','4C696E652039',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'C','8','1000','SWIMMING_LINES',null,null,'SWIMMING_LINE_ID','POOLAPP',null,'10','0','1','10','10','0','1','10','3',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C102','C10B',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XISS_SWIMMING_LINE_ID',null,null,'SWIMMING_LINES','POOLAPP','POOLAPP','10','1','10','1','1','1','1','0','10',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,'16.SWIMMING_LINE_ID',EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'T','8','2','SWIMMING_LINES_SCHEDULES',null,null,null,'POOLAPP',null,'10000','20','8','10000',null,null,null,null,'0',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','1000','SWIMMING_LINES_SCHEDULES',null,null,'SCHEDULE_ID',
'POOLAPP',null,'10000','0,0001','10000','10000','0','14498','24497',
'5',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,'C302
2D63','C3032D62',null,null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','4102','SWIMMING_LINES_SCHEDULES',null,null,'SWIMMING_LI
NE_ID','POOLAPP',null,'10','0,00005','10','10000','0','1','10','3','1',
'10000','10','0',null,to_date('08.05.22','DD.MM.RR'),null,'C102','C10
B','C10B',null,EMPTY_CLOB(),EMPTY_BLOB());
```

```
-- VISITS
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'T','8','2','VISITS',null,null,null,'POOLAPP',null,'10000000','5
8037','38','10000000',null,null,null,null,'0',null,null,null,null,to_d
ate('08.05.22','DD.MM.RR'),null,null,null,null,null,EMPTY_CLOB(),
EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','6','VISITS',null,null,'COACH_ID','POOLAPP',null,'745','
0,00105','745','5599','0','5','1000','4','1','2306','436','5',null,to_
date('08.05.22','DD.MM.RR'),null,'C106','C20B','C20525',null,
EMPTY_CLOB(),EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'C','8','2','VISITS',null,null,'VISIT_ID','POOLAPP',null,'100000
00','0,000000102288186737314','10000000','10000000','0','2932891','127
09190','6',null,null,null,null,null,to_date('08.05.22','DD.MM.RR'),nul
```

```

1, 'C4035E1D5C', 'C40D475C5B', null, null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'VISITS', null, null, 'VISIT_DATE', 'POOLAPP', null, '3649
', '0,000274047684297068', '3649', '10000000', '0', '2455199', '2458847', '8'
, null, null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, '786E010
2010101', '78770C1D010101', null, null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '2', 'VISITS', null, null, 'STATUS', 'POOLAPP', null, '2', '0,5'
, '2', '10000000', '0', '3388445322252520000000000000000000000000', '4170523987
84541000000000000000000000000000', '8', null, null, null, null, null, to_date('08.0
5.22', 'DD.MM.RR'), null, '414253454E54', '50524553454E54', null, null,
EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '1000', 'VISITS', null, null, 'ABONEMENT_ID', 'POOLAPP', null,
'759', '0,00131752305665349', '759', '10000000', '0', '1', '1000', '4', null, n
ull, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, 'C102', 'C20B', n
ull, null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null, 'C', '8', '1000', 'VISITS', null, null, 'CLIENT_ID', 'POOLAPP', null, '97
856', '0,0000102190974493133', '97856', '10000000', '0', '2294', '100000', '5
', null, null, null, null, null, null, to_date('08.05.22', 'DD.MM.RR'), null, 'C2175F
', 'C30A6450', null, null, EMPTY_CLOB(), EMPTY_BLOB());
Insert into poolapp.POOL_STATS_TABLE
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values

```



```
(null,'C','8','4102','VISITS',null,null,'LESSON_ID','POOLAPP',null,'6',
,'0,0000000511440933686569','6','10000000','0','1','6','3','1','100000
00','6','0',null,to_date('08.05.22','DD.MM.RR'),null,'C102','C107','C1
07',null, EMPTY_CLOB(), EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIVS_ABONEMENT_ID',null,null,'VISITS','POOLAPP','PO
OLAPP','10000000','20288','759','26','88','67307','2','10000000',null,
null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null
,null,'12.ABONEMENT_ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIVS_CLIENT_ID',null,null,'VISITS','POOLAPP','POOLA
PP','10000000','21658','96934','1','10','1029265','2','10000000',null,
null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null
,null,'9.CLIENT_ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIVS_COACH_ID',null,null,'VISITS','POOLAPP','POOLAP
P','10000000','20037','745','26','165','123468','2','10000000',null,nu
ll,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,
null,'8.COACH_ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N
9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
```

```
(null,'I','8','2','XIVS_LESSON_ID',null,null,'VISITS','POOLAPP','POOLA
PP','10000000','19058','6','3176','9864','59184','2','10000000',null,n
ull,null,null,null,null,to_date('09.05.22','DD.MM.RR'),null,null,null,null,
null,'9.LESSON_ID', EMPTY_BLOB());
```

```
Insert into poolapp.POOL_STATS_TABLE
```

```
(STATID,TYPE,VERSION,FLAGS,C1,C2,C3,C4,C5,C6,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,D1,T1,R1,R2,R3,CH1,CL1,BL1) values
(null,'I','8','2','XIVS_VISIT_ID',null,null,'VISITS','POOLAPP','POOLAPP',
'10000000','19082','10000000','1','1','74227','2','540000',null,null,
null,null,null,null,to_date('08.05.22','DD.MM.RR'),null,null,null,null,
null,'8.VISIT_ID', EMPTY_BLOB());
```

```
-- delete old state and apply new
```

```
EXEC DBMS_STATS.delete_schema_stats('POOLAPP');
```

```
exec dbms_stats.import_schema_stats(OWNNAME=>'POOLAPP',
```

```
STATTAB=>'POOL_STATS_TABLE');
```