

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«Национальный исследовательский ядерный университет «МИФИ»**

**Факультет «Информационная безопасность»**

Отчёт по

Лабораторной работе №1-3:

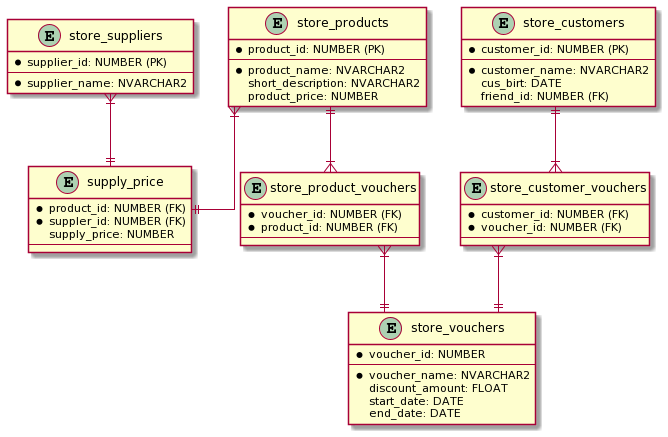
«Сложные запросы на выборку. Соединения»

Выполнил: До Тхуи Хиен

Группа: Б18-565

Преподаватель:  Куприяшин М.А.

Москва 2021



1. **Цель работы**

Приобретение опыта написания более сложных запросов к базе данных Oracle, в частности — с применением аналитических функций и операций соединения.

1. **Выполнение сложных запросов на выборку**
   1. Подзапросы

* Этот запрос вернет список продуктов, у которых цена больше, чем 10:

SELECT\*FROM store\_products WHERE product\_id IN (SELECT product\_id FROM store\_products WHERE product\_price > 10);

* Этот запрос вернет список покупателей, у которых день рождения больше, чем у William:

SELECT\*FROM store\_customers WHERE cus\_birt > (SELECT cus\_birt FROM store\_customers WHERE cus\_n='William');

* 1. Соединение
* Этот запрос позволяет получить список продуктов с названием поставщиков, предлагающих товары:

SELECT product\_name,supplier\_name FROM store\_products p,store\_suppliers s, supply\_price su WHERE p.product\_id = su.product\_id and su.supplier\_id = s.supplier\_id;

* Этот запрос позволяет получить список пар покупателей с одинаковым месцем рождения:

SELECT c1.cus\_n, c2.cus\_n FROM store\_customers c1, store\_customers c2 WHERE

(EXTRACT (month FROM c1.cus\_birt))=(EXTRACT (month FROM c2.cus\_birt)) and c1.cus\_n != c2.cus\_n;

* 1. Иерархические запросы (CONNECT BY, PRIOR, START WITH)

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_id | cus\_n | cus\_birth | friend\_id |
| 1 | Sasha | 21-SEP-70 |  |
| 2 | Mike | 17-APR-98 | 1 |
| 3 | Harry | 04-OCT-65 | 1 |
| 4 | Jamie | 21-DEC-78 | 2 |
| 5 | William | 21-NOV-85 | 2 |
| 6 | Valentina | 20-MAY-97 | 8 |
| 7 | Zac | 31-OCT-80 | 6 |
| 8 | Peter | 13-FEB-70 | 9 |
| 9 | Mai | 19-JAN-98 |  |
| 10 | Lee | 17-SEP-82 | 8 |

Таблица1-Customers

Для выполнения этих запросов, добавили в таблицу Customers один столбец friend\_id. Предположим, что каждый покапатель приглашает своих друзей сделать покупки в магазине и их друзья продолжают приглашать других чтобы получить больше скидка.

* Этот запрос выводит уровень покупателя в иерархии и соответствующий друг:

SELECT level, customer\_id, cus\_n, Prior cus\_n friend\_name

FROM store\_customers

CONNECT BY PRIOR customer\_id=friend\_id

START WITH friend\_id IS NULL;

* Создать путь значения от корневого узла до текущего узла:

SELECT cus\_n, Sys\_Connect\_By\_Path (customer\_id,':') Path

FROM store\_customers

CONNECT BY PRIOR customer\_id=friend\_id

START WITH friend\_id IS null;

* 1. Аналитические функции (OVER)
* Этот запрос позволяет рассчитать общий купон, отсортированные по годам и месяцам:

SELECT EXTRACT (year from start\_date),

EXTRACT (month from start\_date),

SUM (discount\_amount) OVER (ORDER BY (EXTRACT (month from start\_date)), (EXTRACT (year from start\_date))

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as SUM

FROM store\_vouchers;

* Этот запрос позволяет рассчитать минимально цену для импорта продуктов:

SELECT product\_id,supplier\_id, supply\_price,

MIN (supply\_price) OVER (PARTITION BY product\_id) AS min\_price

FROM supply\_price;

* Этот запрос позволяет рассчитать количество продуктов, которые поставщик может поставить:

SELECT supplier\_id,

COUNT (\*) OVER (PARTITION BY product\_id) AS amount\_product

FROM supply\_price;

2.5 Теоретико-множественные операции (UNION, UNION ALL, INTERSECT, MINUS)

* Получить разность множеств списка customer\_id из двух таблиц store\_customers и store\_customer\_vouchers:

SELECT customer\_id FROM store\_customers

MINUS

SELECT customer\_id FROM store\_customer\_vouchers;

* Получить пересечение списка product\_id из двух таблиц store\_products и store\_product\_vouchers:

SELECT product\_id FROM store\_products

INTERSECT

SELECT product\_id FROM store\_product\_vouchers;

* Получить объединение списка voucher\_id из двух таблиц store\_vouchers и store\_customer\_vouchers и устранить возникающие при объединении дубликаты:

SELECT voucher\_id FROM store\_vouchers

UNION

SELECT voucher\_id FROM store\_customer\_vouchers;

* Получить объединение списка voucher\_id из двух таблиц store\_vouchers и store\_customer\_vouchers но не устранить возникающие при объединении дубликаты:

SELECT voucher\_id FROM store\_vouchers

UNION ALL

SELECT voucher\_id FROM store\_customer\_vouchers;

1. **Залючение**

Целью данного исследования было приобретение опыта написания более сложных запросов к базе данных Oracle, в частности — с применением аналитических функций и операций соединения.

В данной работе разработали некоторые различные запросы с применением различных возможностей, предоставляемых стандартом SQL и Oracle Database.

Ссылка на приложения:

<https://github.com/ThuyHienDo/Oracle-SQL/blob/main/lab1-3.txt>