Университет ИТМО

Кафедра ВТ

Системы баз данных

Лабораторная работа.

Часть 1.

Работу выполнил:

Студент 3 курса

Группы P3315

Назарьев Сергей

Санкт-Петербург

2015 г.

-- #1

--- Вывести содержимое всех столбцов таблицы Н\_СЕССИЯ, поля типа DATE

--- представить в формате DD.MM.YYYY HH24:MI:SS

**SELECT** ИД, СЭС\_ИД, ЧЛВК\_ИД, АУДИТОРИЯ,

 TO\_CHAR(ДАТА, 'DD.MM.YYYY HH24:MI:SS') **AS** ДАТА,

 TO\_CHAR(ВРЕМЯ, 'DD.MM.YYYY HH24:MI:SS') **AS** ВРЕМЯ,

 TO\_CHAR(КОГДА\_СОЗДАЛ, 'DD.MM.YYYY HH24:MI:SS') **AS** КОГДА\_СОЗДАЛ,

 TO\_CHAR(КОГДА\_ИЗМЕНИЛ, 'DD.MM.YYYY HH24:MI:SS') **AS** КОГДА\_ИЗМЕНИЛ,

 TO\_CHAR(ВРЕМЯ\_К, 'DD.MM.YYYY HH24:MI:SS') **AS** ВРЕМЯ\_К,

 TO\_CHAR(ДАТА\_К, 'DD.MM.YYYY HH24:MI:SS') **AS** ДАТА\_К,

 АУДИТОРИЯ\_К, УЧГОД, ГРУППА, СЕМЕСТР, КТО\_СОЗДАЛ, КТО\_ИЗМЕНИЛ

**FROM** Н\_СЕССИЯ;

-- #2

--- Вывести неповторяющиеся названия дисциплин из таблицы Н\_ДИСЦИПЛИНЫ

**SELECT** **DISTINCT** НАИМЕНОВАНИЕ **FROM** Н\_ДИСЦИПЛИНЫ;

-- #3

--- Вывести округленное значение разницы между днем рождения произвольной

--- персоны из Н\_ЛЮДИ и началом текущего учебного года.

**SELECT** ROUND(TO\_DATE('2014/09/01', 'yyyy/mm/dd') - ДАТА\_РОЖДЕНИЯ)

**FROM** Н\_ЛЮДИ

**WHERE** ИД = (**SELECT** **MIN**(ИД) **FROM** Н\_ЛЮДИ);

-- #4

--- Получить фамилии и инициалы людей (в виде «Иванов И.И.), которые

--- родились в том же месяце, что и произвольная персона из Н\_ЛЮДИ.

**SELECT** ФАМИЛИЯ || ' ' || SUBSTR( ИМЯ, 0, 1) || '.' ||

 SUBSTR( ОТЧЕСТВО, 0, 1) || '.'

**FROM** Н\_ЛЮДИ

**WHERE** TO\_CHAR(ДАТА\_РОЖДЕНИЯ, 'MM') = (**SELECT** TO\_CHAR(ДАТА\_РОЖДЕНИЯ, 'MM')

 **FROM** Н\_ЛЮДИ

 **WHERE** ИД = 110103 **AND**

 ФАМИЛИЯ = 'Афанасьев');

-- #5

--- Вывести фамилии, имена, отчества и номера (ИД) людей, фамилии которых

--- начинаются на те же 2 буквы, что и у произвольной персоны из Н\_ЛЮДИ,

--- упорядочить по убыванию ФИО, вывести не более 75 строк.

**SELECT** ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

**FROM** Н\_ЛЮДИ

**WHERE** SUBSTR(ФАМИЛИЯ, 0, 2) = (**SELECT** SUBSTR(ФАМИЛИЯ, 0, 2)

 **FROM** Н\_ЛЮДИ

 **WHERE** ИД = 110103 **AND**

 ФАМИЛИЯ = 'Афанасьев') **AND**

 ROWNUM <= 75

**ORDER** **BY** Фамилия **DESC**;

-- #6

--- Вывести список персон (фамилия, имя, отчество, ИД), для которых инициалы не

--- равны «А», «Б», «З» и «К», «У». Представить не менее 2х вариантов запроса.

**SELECT** ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

**FROM** Н\_ЛЮДИ

**WHERE** **NOT** REGEXP\_LIKE (SUBSTR( ИМЯ, 0, 1), '^[АБЗКУ]{1}$') **AND**

 **NOT** REGEXP\_LIKE (SUBSTR( ОТЧЕСТВО, 0, 1), '^[АБЗКУ]{1}$');

**SELECT** ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ИД

**FROM** Н\_ЛЮДИ

**WHERE** SUBSTR( ИМЯ, 0, 1) **NOT** **IN** ('А', 'Б', 'З', 'К', 'У') **AND**

 SUBSTR( ОТЧЕСТВО, 0, 1) **NOT** **IN** ('А', 'Б', 'З', 'К', 'У');

-- #7

--- Вычислите количество персон, имя которых такое же,

--- как у произвольной персоны из Н\_ЛЮДИ, учесть возможность

--- наличия дубликатов.

**SELECT** **COUNT**(**DISTINCT** ИД)

**FROM** Н\_ЛЮДИ

**WHERE** ИМЯ = (**SELECT** ИМЯ **FROM** Н\_ЛЮДИ **WHERE** ИД = 110103);

-- #8

--- Вывести, используя таблицу Н\_ВЕДОМОСТИ, удвоенные

--- (значение оценки\*2) оценки произвольной персоны из Н\_ЛЮДИ.

--- Использовать NOT IN и регулярные выражения.

**SELECT** TO\_NUMBER(ОЦЕНКА)\*2 **AS** УДВОЕННАЯ\_ОЦЕНКА **FROM** Н\_ВЕДОМОСТИ

**WHERE** ЧЛВК\_ИД = (**SELECT** ИД

 **FROM** Н\_ЛЮДИ

 **WHERE** **NOT** ИД **NOT** **IN** (110103) **AND**

 REGEXP\_LIKE(ФАМИЛИЯ, '^Афанасьев$')) **AND**

 REGEXP\_LIKE(ОЦЕНКА, '^[0-9]{1}$')

-- #9

--- Вывести, используя таблицу Н\_ВЕДОМОСТИ, сумму оценок 7и

--- произвольных персон из Н\_ЛЮДИ.

--- 7 персон задаются условием к таблице Н\_ЛЮДИ.

**SELECT** **SUM**(TO\_NUMBER(ОЦЕНКА)) **FROM** Н\_ВЕДОМОСТИ

**WHERE** ЧЛВК\_ИД **IN** (**SELECT** ИД

 **FROM** Н\_ЛЮДИ

 **WHERE** ИД **IN** (110103, 110105, 110106, 110107, 110108,

 110104, 110109)) **AND**

 ОЦЕНКА **NOT** **IN** ('зачет');

-- #10

--- Получить декартово произведение N таблиц, где N равно 3й цифре вашего

--- табельного номера.

**SELECT** \* **FROM** Н\_ЛЮДИ, Н\_ВЕДОМОСТИ;

-- #11

--- Вывести, используя таблицу Н\_ВЕДОМОСТИ, среднюю оценку 7и

--- произвольных персон из Н\_ЛЮДИ, их фамилии, имена и отчества.

**SELECT** AVG(TO\_NUMBER(ОЦЕНКА)), ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО

**FROM** Н\_ВЕДОМОСТИ

**JOIN** Н\_ЛЮДИ **ON** Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

**WHERE** ЧЛВК\_ИД **IN** (**SELECT** ИД

 **FROM** Н\_ЛЮДИ

 **WHERE** ИД **IN** (110103, 110105, 110106, 110107, 110108,

 110104, 110109)) **AND**

 REGEXP\_LIKE(ОЦЕНКА, '^[0-9]{1}$')

**GROUP** **BY** ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО;

-- #12

--- Сформировать SQL-запрос для получения таблицы вида:

--- +====================================================================+

--- | | Средняя оценка | Количество |

--- | | | оценок |

--- +====================================================================+

--- | Оценки 4 и 5 во всем университете | 4.3 | 98 |

--- +--------------------------------------------------------------------+

--- | Оценки «зачет» в произвольном | - | 86 |

--- | учебном году во всем университете | | |

--- +--------------------------------------------------------------------+

--- | Расстояние Левенштайна до вашей | 2 | - |

--- | фамилии от фамилии 10 персон, | | |

--- | имеющих оценки 3, 4 и 5 | | |

--- +--------------------------------------------------------------------+

**SELECT** '', 'Средняя оценка', 'Количество оценок'

**FROM** DUAL

**WHERE** 0=1

**UNION** **SELECT** 'Оценки 4 и 5 во всем университете',

 **REPLACE**(TO\_CHAR(ROUND(AVG(ОЦЕНКА),1)),',','.'),

 TO\_CHAR(**COUNT**(ОЦЕНКА))

 **FROM** Н\_ВЕДОМОСТИ

 **WHERE** ОЦЕНКА **IN** ('4','5')

 **GROUP** **BY** ''

**UNION** **SELECT** 'Оценки «зачет» в произвольном учебном году во всем университете',

 '-', TO\_CHAR(**COUNT**(ОЦЕНКА))

 **FROM** Н\_ВЕДОМОСТИ

 **WHERE** ОЦЕНКА = 'зачет' **AND**

 TO\_CHAR(ДАТА, 'YYYY') = '2003'

 **GROUP** **BY** ''

**UNION** **ALL** **SELECT** 'Расстояние Левенштайна до вашей фамилии от фамилии 10 персон,

имеющих оценки 3, 4 и 5',

 TO\_CHAR(utl\_match.edit\_distance('Назарьев', ФАМИЛИЯ)), '-'

**FROM** (**SELECT** ФАМИЛИЯ

 **FROM** Н\_ЛЮДИ

 **WHERE** ИД **IN** (

 **SELECT** ЧЛВК\_ИД

 **FROM** Н\_ВЕДОМОСТИ

 **WHERE** ОЦЕНКА **IN** ('3','4','5') **AND**

 ROWNUM <=10));

-- #13

--- Получить список персон, получивших оценки 3 и 4 с 01.09 прошлого

--- календарного года по 20.07 текущего календарного года, упорядочить

--- список по ФИО. Использование объединений таблиц запрещено.

**SELECT** ФАМИЛИЯ || ' ' || ИМЯ || ' ' || ОТЧЕСТВО **AS** ФИО

**FROM** Н\_ЛЮДИ

**WHERE** ИД **IN** (

 **SELECT** **DISTINCT** ЧЛВК\_ИД

 **FROM** Н\_ВЕДОМОСТИ

 **WHERE** ОЦЕНКА **IN**('3','4') **AND**

 ДАТА **BETWEEN**

 TO\_DATE('01.09.' || (TO\_NUMBER(TO\_CHAR(**CURRENT\_DATE**, 'YYYY')-5)))

 **AND**

 TO\_DATE('20.07.' || TO\_NUMBER(TO\_CHAR(**CURRENT\_DATE**, 'YYYY'))))

**ORDER** **BY** 1

-- #14

--- Получить список людей с наиболее частыми сочетаниями фамилии, имени и

--- отчества, сумма оценок которых не превышает сумму цифр ИД произвольной

--- персоны из таблицы Н\_ЛЮДИ.

**SELECT** ЧЛВК\_ИД **FROM**

 (**SELECT** ЧЛВК\_ИД, **SUM**(ОЦЕНКА) **AS** СУММА

 **FROM** Н\_ВЕДОМОСТИ

 **WHERE** ЧЛВК\_ИД **IN**

 (**SELECT** ИД

 **FROM** Н\_ЛЮДИ

 **WHERE** ФАМИЛИЯ || ' ' || ИМЯ || ' ' || ОТЧЕСТВО **IN** (

 **SELECT** ФИО **FROM** (

 **SELECT** ФАМИЛИЯ || ' ' || ИМЯ || ' ' || ОТЧЕСТВО

 **AS** ФИО,

 **COUNT**(1) **AS** CNT

 **FROM** Н\_ЛЮДИ

 **GROUP** **BY** 1

 **ORDER** **BY** 2 **DESC**)

 **WHERE** CNT = (

 **SELECT** **MAX**(**COUNT**(ФАМИЛИЯ || ' ' || ИМЯ || ' ' || ОТЧЕСТВО))

 **FROM** Н\_ЛЮДИ

 **GROUP** **BY** ФАМИЛИЯ || ' ' || ИМЯ || ' ' || ОТЧЕСТВО)))

 **AND** REGEXP\_LIKE(ОЦЕНКА, '^[0-9]{1}$')

 **GROUP** **BY** ЧЛВК\_ИД)

**WHERE** СУММА <= (

 **WITH** t **AS** (**SELECT** **MAX**(ИД) **AS** a **FROM** Н\_ЛЮДИ)

 **SELECT** 9 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^9]', ''))),0)

 + 8 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^8]', ''))),0)

 + 7 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^7]', ''))),0)

 + 6 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^6]', ''))),0)

 + 5 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^5]', ''))),0)

 + 4 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^4]', ''))),0)

 + 3 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^3]', ''))),0)

 + 2 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^2]', ''))),0)

 + 1 \* **COALESCE**((**LENGTH**(REGEXP\_REPLACE(a, '[^1]', ''))),0)

 **FROM** t);