**Міністерство освіти і науки України**

**Харківський національний університет радіоелектроніки**

**Кафедра штучного інтелекту**

Звіт з лабораторної роботи №2

курс: Організація баз даних та знань

з теми: “Використання функцій Oracle для створення запитів до бази даних. Використання підзапитів ”

Виконала:

студентка гр. ІТКНу-18-1

Щетініна Анастасія Володимирівна

Перевірили:

Магдаліна Ігор Валерійович

Прийма Анна Константинівна

Харків 2018

1. Из строки ««Intelligent Decision Support Systems» с помощью строковых функций получить ««Intelligent Systems».

**SELECT** **SUBSTR**('Intelligent Desision Support Systems',1,11)

||' '

|| **SUBSTR**('Intelligent Desision Support Systems',-7,7)

**AS** ExtractString

**FROM** DUAL



1. В строке ««initializations» выделить комбинацию, где встречаются символы ««i» и ««t» попарно (порядок учитывать). Вывести номер позиции в строке.

**SELECT** **INSTR**('initializations','it')

**AS** ExtractString

**FROM** DUAL



1. С помощью служебной таблицы узнать текущую дату. Вывести дату в различных форматах: отображение месяца как числовое значение, отображение месяца как сокращенное название, отображение полного названия месяца

**SELECT** **TO\_CHAR**(**sysdate**,'dd mm yyyy')"Today" **FROM** dual;



**SELECT** **TO\_CHAR**(**sysdate**,'dd mon yyyy')"Today" **FROM** dual;



**SELECT** **TO\_CHAR**(**sysdate**,'dd Month yyyy')"Today" **FROM** dual;



1. Вывести дату полностью, текущий год, месяц из установленной вами даты.

**select** **TO\_DATE**('31 12 1998', 'dd mm yyyy') **as** dateOfBirth **from** dual;

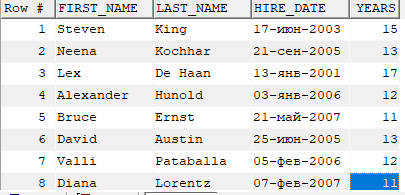


1. Указать сотрудников, у которых стаж работы на сегодняшний день не менее 10 лет. Вывести имя, фамилию сотрудника, стаж (только годы).

**select** employees.first\_name, employees.last\_name, employees.hire\_date,**FLOOR**((**sysdate** - employees.hire\_date)/365 ) years

**from** hr.employees

**where** **FLOOR**(**MONTHS\_BETWEEN**(**sysdate**, employees.hire\_date)/12 )>=10;



1. Вывести среднюю зарплату сотрудников каждого отдела. Поле «Средняя зарплата» должно содержать корректно округленные значения. Отделы должны быть отсортированы в алфавитном порядке.

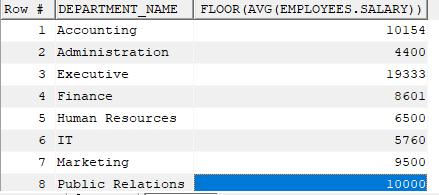
**select** departments.department\_name, **FLOOR**(**AVG**(employees.salary))

**from** hr.departments, hr.employees

**where** employees.department\_id = departments.department\_id

**group** **by** departments.department\_name

**order** **by** departments.department\_name **asc**;



1. Указать количество работающих сотрудников.

**select** **count**(employees.employee\_id) countOfEmployees

**from** hr.employees;

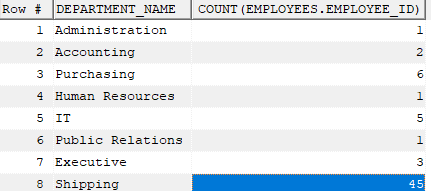


1. Указать число сотрудников в каждом отделе.

**select** departments.department\_name, **count**(employees.employee\_id)

**from** hr.employees, hr.departments

**where** employees.department\_id = departments.department\_id

**group** **by** departments.department\_name;

1. Проверить, существуют ли сотрудники, у которых не указан

отдел.

**select** employees.first\_name, employees.last\_name, employees.department\_id

**from** hr.employees

**where** **NVL**(employees.department\_id,0)=0;

1. Какое количество денежных средств необходимо выделять ежемесячно, чтобы все сотрудники в каждом отделе получали зарплату. Указать отдел, необходимое количество денежных средств.

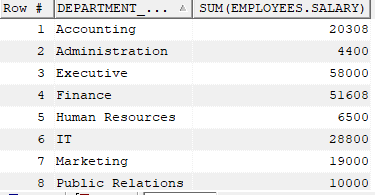
**select** departments.department\_name, **sum**(employees.salary)

**from** hr.departments, hr.employees

**where** employees.department\_id = departments.department\_id

**GROUP** **BY** departments.department\_name

**order** **by** departments.department\_name **asc**;



11. Найти всех сотрудников, которые получают премию. Результирующая таблица должна содержать такие поля:

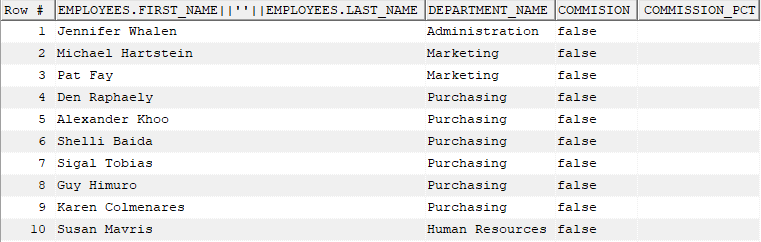
|  |  |  |
| --- | --- | --- |
| Ф.И.О | Информация о премии | Отдел |
|  | (true/false) |  |

**select** employees.first\_name||' '||employees.last\_name,departments.department\_name,

**decode**(employees.commission\_pct, **null**, 'false', 'true') commision,employees.commission\_pct

**from** hr.employees, hr.departments

**where** employees.department\_id = departments.department\_id;



1. Вывести количество сотрудников, у которых средняя зарплата за полгода превышает 50000.

**select** **count**(employees.employee\_id)

**from** hr.employees

**where** employees.salary\*6>50000;

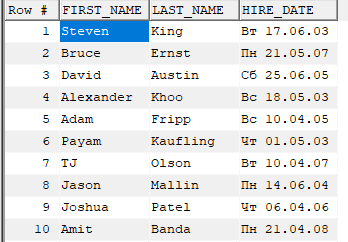


1. Найти всех сотрудников, которые были зачислены на работу в период 2 квартала 2003 года.

**select** employees.first\_name, employees.last\_name, employees.hire\_date

**from** hr.employees

**where** **extract**(**month** **from** employees.hire\_date) **in** (4,5,6)

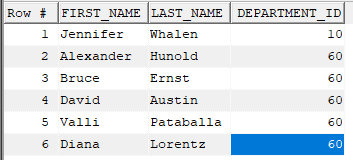


14. Написать 2 варианта запроса, который находит сотрудников, работающих в 10, 60, 240, 210 отделах.

**select** employees.first\_name, employees.last\_name, employees.department\_id

**from** hr.employees

**where** employees.department\_id **in** (10,60,240,210);

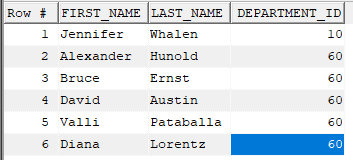


**select** employees.first\_name, employees.last\_name, employees.department\_id

**from** hr.employees

**where** employees.department\_id=10 **or** employees.department\_id=60

**or** employees.department\_id=210 **or** employees.department\_id=240;



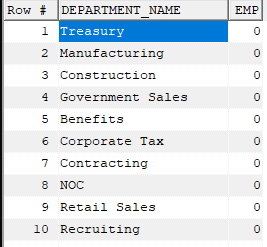
1. Найти отдел, в котором нет сотрудников.

**select** departments.department\_name, **count**(employees.employee\_id) emp **from** hr.departments **left** **outer** **join** hr.employees

**on** employees.department\_id=departments.department\_id

**having** **count**(employees.employee\_id)=0

**group** **by** departments.department\_name



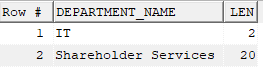
1. Определить, в наименовании какого отдела содержится наибольшее/наименьшее количество символов. Вывести название отдела, количество символов. Информацию о наибольшем/наименьшем количестве символов вывести одним запросом.

**select** departments.department\_name, **LENGTH**(departments.department\_name) len

**from** hr.departments

**where** **LENGTH**(departments.department\_name)=(**select** **max**(**LENGTH**(departments.department\_name))**from** hr.departments)

**or** **LENGTH**(departments.department\_name)=(**select** **min**(**LENGTH**(departments.department\_name))**from** hr.departments) ;



1. Найти отдел, в котором средняя заработная плата больше, чем в других отделах. Вывести название отдела, уровень зарплаты

**select** \* **from** (**select** departments.department\_name, **FLOOR**(**AVG**(employees.salary)) salary

**from** hr.departments, hr.employees

**where** employees.department\_id = departments.department\_id

**group** **by** departments.department\_name

**order** **by** **FLOOR**(**AVG**(employees.salary)) desc )

**where** **rownum**=1;



1. С помощью SQL-запроса определить количество пробелов в строке «Я являюсь студентом факультета КИУ».

**select** **LENGTH**(**REPLACE**('Я являюсь студентом КИУ', ' ', ' ')) - **LENGTH**('Я являюсь студентом КИУ')len

**from** dual;



1. Выполнить запрос и получить данные о сотруднике с выбранным вами employee\_id.

**select** employees.first\_name||' '||employees.last\_name "Фамилия имя" , jobs.job\_title Должность,

employees.phone\_number "Номер телефона", departments.department\_name Отдел

**from** hr.employees, hr.departments, hr.jobs

**where** employees.job\_id=jobs.job\_id **and** employees.department\_id=departments.department\_id **and** employees.employee\_id=101;

1. Найти отдел, в котором премию получают наибольшее количество сотрудников Вывести количество сотрудников и название отдела.

**select** departments.department\_name, **count** (employees.commission\_pct) comm\_pct

**from** hr.employees **join** hr.departments **on** employees.department\_id=departments.department\_id

**having** **count**(employees.employee\_id)=(**select** **max**(**count**(employees.employee\_id)) **from** hr.employees

**where** employees.commission\_pct **is** **not** **null**

**group** **by** employees.department\_id)

**group** **by** departments.department\_name;



1. Найти сотрудников, у которых непосредственные руководители работают в отделе, отличном от того, в котором работает сам сотрудник. Итоговая выборка должна отображать: ГО сотрудника, наименование отдела, в котором этот сотрудник работает, ГО непосредственного руководителя, наименование отдела, в котором работает руководитель.

**select** employees.employee\_id, departments.department\_name,e.manager\_id, d.department\_name

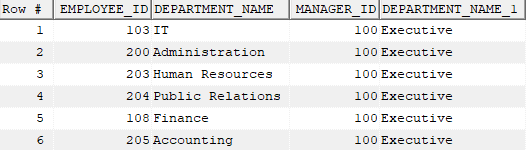
**from** hr.employees, hr.departments, departments d, employees e

**where** employees.manager\_id=e.employee\_id

**and** departments.department\_id!=d.department\_id

**and** employees.department\_id=departments.department\_id

**and** e.department\_id=d.department\_id



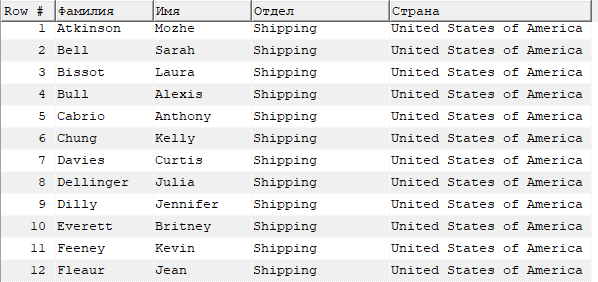
1. Вывести список сотрудников, полные названия соответствующих отделов, название страны. Список сгруппировать по отделам, причем фамилии сотрудников в рамках каждого отдела должны быть отсортированы в алфавитном порядке по возрастанию, а названия отделов - по убыванию.

**select** employees.last\_name "Фамилия",employees.first\_name "Имя", departments.department\_name "Отдел", countries.country\_name "Страна"

**from** hr.employees, hr.departments, hr.countries, hr.locations

**where** employees.department\_id=departments.department\_id **and** departments.location\_id=locations.location\_id **and** locations.country\_id=countries.country\_id

**order** **by** departments.department\_name desc, employees.last\_name **asc**;



1. Найти сотрудников, которые получают наибольшую зарплату по своей должности

**select** jobs.job\_title, employees.salary, employees.last\_name||' '||employees.first\_name employee

**from** hr.employees, hr.jobs

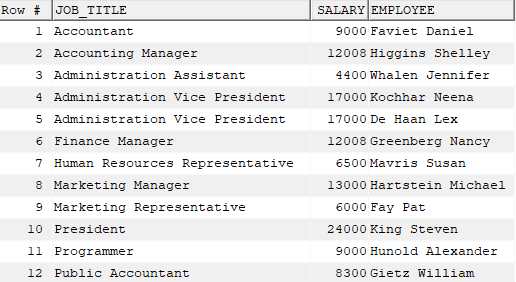
**where** employees.job\_id=jobs.job\_id

**and** (employees.job\_id, employees.salary) **in** (**select** employees.job\_id, **max**(employees.salary)

**from** hr.employees

**group** **by** employees.job\_id)

**order** **by** jobs.job\_title **asc**;



Вывод: в ходе лабораторной были использованы функции для работы с датой - MONTHS\_BETWEEN,со строками – SUBSTR, INSTR, TO\_CHAR, LENGTH, COUNT, агрегатные – AVG, SUM, MIN, MAX, округления – FLOOR; оба вида сортировок (desc, asc), группировки и оператор HAVING для использования агрегатных функций для условия выборки.

Был выучен новый оператор DECODE, декодирует выражение способом, подобным логике IF-THEN-ELSE, которая используется в различных языках. Функция DECODE декодирует выражение после сравнения его с каждым поисковым значением. Если выражение является тем же самым, как поисковое, результат возвращается. Если значение по умолчанию опускается, значение NULL возвращается, когда поисковое значение не соответствует ни одному из значений результата.

Были выполнены сложные запросы с использованием подзапросов, а также оператора JOIN (LEFT OUTER JOIN, RIGHT OUTER JOIN), который является реализацией [операции соединения](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_(%D0%A1%D0%A3%D0%91%D0%94)) [реляционной алгебры](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0), предназначенный для обеспечения выборки данных из двух таблиц и включения этих данных в один результирующий набор