## Лабораторная работа 1

* Выборка данных из разных таблиц
* Описание структуры таблиц
* Выполнение арифметических вычислений и задание имён столбцов
  1. Команда SELECT содержит 4 ошибки. Укажите их.

|  |  |
| --- | --- |
| SELECT employee\_id, last\_name,  salary \* 12 "ANNUAL SALARY"  FROM employees; | SELECT employee\_id, last\_name sal x 12 ANNUAL SALARY FROM employees; |

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| DEPARTMENT\_ID | NOT NULL | NUMBER(4) |
| DEPARTMENT\_NAME | NOT NULL | VARCHAR(2) |
| MANAGER\_ID |  | NUMBER(6) |
| LOCATION\_ID |  | NUMBER(4) |

* 1. Покажите структуру таблицы DEPARTMENS. Выберите данные из неё.

DESCRIBE departments;

SELECT \*

FROM departments;

* 1. Покажите структуру таблицы EMPLOYEES.

DESCRIBE employees;

* 1. Составьте запрос для вывода имени каждого служащего, должности, даты найма и номера. Номер служащего должен быть первым.

SELECT employee\_id, first\_name, job\_id, hire\_date

FROM employees

….

|  |
| --- |
| JOB\_ID |
| AC\_ACCOUNT |
| AC\_MGR |
| AD\_ASST |
| AD\_PRES |
| AD\_VP |
| IT\_PROG |
| MK\_MAN |
| MK\_REP |
| SA\_MAN |
| SA\_REP |
| ST\_CLERK |
| ST\_MAN |

* 1. Составьте запрос для вывода неповторяющихся должностей из таблицы EMPLOYEES.

SELECT distinct job\_id

FROM employees

* 1. Выведите на экран имя, соединённое с идентификатором должности через запятую и пробел. Назовите новый столбец Employee and Title

SELECT first\_name||', '||job\_id "Employee and Title"

FROM employees

|  |
| --- |
| Employee and Title |
| King, AD\_PRES |
| Kochhar, AD\_VP |
| De Haan, AD\_VP |
| Hunold, IT\_PROG |
| Ernst, IT\_PROG |
| Lorentz, IT\_PROG |
| Mourgos, ST\_MAN |
| Rajs, ST\_CLERK |
| Davies, ST\_CLERK |
| Matos, ST\_CLERK |
| Vargas, ST\_CLERK |

* 1. Создайте запрос для вывода всех данных из таблицы EMPLOYEES. Разделите столбцы запятыми. Назовите столбец THE\_OUTPUT

SELECT

employee\_id||','||

first\_name||','||

last\_name||','||

email||','||

phone\_number||','||

hire\_date||','||

job\_id||','||

salary||','||

commission\_pct||','||

manager\_id||','||

department\_id

"THE OUTPUT"

FROM employees

|  |
| --- |
| THE\_OUTPUT |
| 100,Steven,King,SKING,515.123.4567,AD\_PRES,17-JUN-87,24000,,90 |
| 101,Neena,Kochhar,NKOCHHAR,515.123.4568,AD\_VP,100,21-SEP-89,17000,,90 |
| 102,Lex,De Haan,LDEHAAN,515.123.4569,AD\_VP,100,13-JAN-93,17000,,90 |
| 103,Alexander,Hunold,AHUNOLD,590.423.4567,IT\_PROG,102,03-JAN-90,9000,,60 |
| 104,Bruce,Ernst,BERNST,590.423.4568,IT\_PROG,103,21-MAY-91,6000,,60 |
| 107,Diana,LorentzDLORENTZ,590.423.5567,IT\_PROG,103,07-FEB-99,4200,,60  ….. |

## Лабораторная работа 2

* Выборка данных и изменение последовательности вывода строк
* Ограничение количества возвращаемых строк с помощью предложения WHERE
* Сортировка строк с помощью предложения ORDER BY

1. Создайте запрос для вывода имени и заработной платы служащих, зарабатывающих более $12000. Сохраните команду SQL в файле lab2\_1.sql. выполните запрос.

|  |  |
| --- | --- |
| LAST\_NAME | SALARY |
| King | 24000 |
| Kochhar | 17000 |
| De Haan | 17000 |
| Hartstein | 13000 |

SELECT last\_name, salary

FROM employees

WHERE salary>12000

1. Создайте запрос для вывода фамилии и номера отдела служащего под номером 176

|  |  |
| --- | --- |
| LAST\_NAME | DEPARTMENT\_ID |
| Taylor | 80 |

SELECT last\_name, department\_id

FROM employees

WHERE department\_id=176

1. Измените файл lab2\_1.sql и выведите фамилии и оклады всех служащих, чей оклад не входит в диапазон от $5000 до $12000. сохраните команду SQL в файле lab2\_3.sql.

|  |  |
| --- | --- |
| LAST\_NAME | SALARY |
| King | 24000 |
| Kochhar | 17000 |
| De Haan | 17000 |
| Lorentz | 4200 |
| Rajs | 3500 |
| Davies | 3100 |
| Matos | 2600 |
| Vargas | 2500 |
| Whalen | 4400 |
| Hartstein | 13000 |

SELECT last\_name, salary

FROM employees

WHERE salary not between 5000 and 12000

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | JOB\_ID | HIRE\_DATE |
| Matos | ST\_CLERK | 15-MAR-98 |
| Taylor | SA\_REP | 24-MAR-98 |

1. Выведите фамилию, идентификатор должности и дату начала работы всех служащих, нанятых в период с 20 февраля 1998 по 1 мая 1998 г. Отсортируйте данные в порядке возрастания даты найма.

SELECT last\_name, job\_id, hire\_date

FROM employees

WHERE hire\_date between ’20.02.98’ and ’01.05.98’

ORDER BY hire\_date

1. Выведите фамилию и номер отдела всех служащих из отделов 20 и 50. Отсортируйте данные по фамилиям в алфавитном порядке.

|  |  |
| --- | --- |
| LAST\_NAME | DEPARTMENT\_ID |
| Davies | 50 |
| Fay | 20 |
| Hartstein | 20 |
| Matos | 50 |
| Mourgos | 50 |
| Rajs | 50 |
| Vargas | 50 |

SELECT last\_name, department\_id

FROM employees

WHERE department\_id=20 or department\_id=50

ORDEER BY last\_name

1. Измените файл lab2\_3.sql для вывода фамилий и окладов служащих отделов 20 и 50, зарабатывающих от $5000 до $12000. Назовите столбцы Employee и Mounthly Salary, соответственно. Вновь сохраните команду SQL в файле lab2\_6.sql. Выполните запрос.

|  |  |
| --- | --- |
| EMPLOYEE | Mounthly Salary |
| Mourgos | 5800 |
| Fay | 6000 |

SELECT last\_name “Employee”, salary “Mounthly Salary”

FROM employees

WHERE department\_id=20 or department\_id=50

and salary not between 5000 and 12000

1. Выведите фамилию и дату найма всех служащих, нанятых в 1994 г.

|  |  |
| --- | --- |
| LAST\_NAME | HIRE\_DATE |
| Higgins | 07-JUN-94 |
| Gietz | 07-JUN-94 |

SELECT last\_name, hire\_date

FROM employees

WHERE hire\_date like ‘%94’

1. Выведите фамилии и должности всех служащих, не имеющих менеджера.

|  |  |
| --- | --- |
| LAST\_NAME | JOB\_ID |
| King | AD\_PRES |

SELECT last\_name, job\_id

FROM employees

WHERE manager\_id is NOT NULL

1. Выведите фамилию, и комиссионные всех служащих, зарабатывающих комиссионные. Отсортируйте данные в порядке убывания окладов и комиссионных.

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | SALARY | COMMISSION\_PCT |
| Abel | 11000 | .3 |
| Zlotkey | 10500 | .2 |
| Taylor | 8600 | .2 |
| Grant | 7000 | .15 |

SELECT last\_name, salary, commission\_pct

FROM employees

WHERE commission\_pct is NOT NULL

ORDER BY salary DESC, commission\_pct DESC

1. Выведите все фамилии служащих, в которых третья буква – *а.*

|  |
| --- |
| LAST\_NAME |
| Grant |
| Whalen |

SELECT last\_name

FROM employees

WHERE last\_name like ‘\_\_a%’

1. Выведите фамилии всех служащих, в которых есть буквы «*а*» и «*е*».

SELECT last\_name

FROM employees

WHERE last\_name like '%a%' and last\_name like '%e%'

|  |
| --- |
| LAST\_NAME |
| De Haan |
| Davies |
| Whalen |
| Hartstein |

1. запросите фамилии, должности и оклады всех служащих, работающих торговыми представителями (SA\_REP) или клерками на складе (ST\_CLERK) и с окладом, не равным $2500, $3500 и $7000.

SELECT last\_name, job\_id, salary

FROM employees

WHERE (job\_id in ('SA\_REP', 'ST\_CLERK'))

and (salary not in (2500, 3500, 7000))

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | JOB\_ID | SALARY |
| Davies | ST\_CLERK | 3100 |
| Matos | ST\_CLERK | 2600 |
| Abel | SA\_REP | 11000 |
| Taylor | SA\_REP | 8600 |

1. Измените lab2\_6.sql так, чтобы получить фамилии оклады и комиссионные всех служащих, у которых сумма комиссионных превышает 20%. Выполните запрос ещё раз. Сохраните запрос в файле lab2\_13.sql.

SELECT last\_name “Employee”, salary “Monthly Salary”, commission\_pct

FROM employees

WHERE commission\_pct>0.2

|  |  |  |
| --- | --- | --- |
| Employee | Monthly Salary | COMMISSION\_PCT |
| Zlotkey | 10500 | .2 |
| Taylor | 8600 | .2 |

## Лабораторная работа 3

* Составление запроса для вывода текущей даты.
* Составление запросов, требующих использования числовых, символьных функций и функций для работы с датами.
* Вычисление продолжительности работы служащего в месяцах и годах.

1. Напишите запрос для вывода текущей даты. Назовите столбец Date.

SELECT current\_date “Date”

FROM dual

|  |
| --- |
| DATE |
| 08-MAR-01 |

1. Выведите номер служащего, его фамилию, оклад и новый оклад, повышенный на 15% и округлённый до целого. Назовите столбец New Salary. Сохраните команду SQL в текстовом файле lab3\_2sql.

SELECT employee\_id, last\_name, salary, round(salary\*1.15, 0) “New Salary”

FROM employees

1. Выполните запрос из файла lab3\_2sql.

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | LAST\_NAME | SALARY | New Salary |
| 100 | King | 24000 | 27600 |
| 101 | Kochhar | 17000 | 19550 |
| 102 | De Haan | 17000 | 19550 |
| 103 | Hunold | 9000 | 10350 |
| 104 | Ernst | 6000 | 6900 |

………….

20 rows selected

SELECT employee\_id, last\_name, salary, round(salary\*1.15, 0) “New Salary”

FROM employees

1. Измените lab3\_2.sql. добавьте ещё один столбец, который будет содержать результат вычитания старого оклада из нового. Назовите столбец Increase. Сохраните изменённый запрос в lab3\_4.sql. Выполните запрос ещё раз.

SELECT employee\_id, last\_name, salary,

round(salary\*1.15, 0) “New Salary”,

ronnd(salaty\*1.15-salary) “Increase”

FROM employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | LAST\_NAME | SALARY | New Salary | Increase |
| 100 | King | 24000 | 27600 | 3600 |
| 101 | Kochhar | 17000 | 19550 | 2550 |
| 102 | De Haan | 17000 | 19550 | 2550 |
| 103 | Hunold | 9000 | 10350 | 1350 |
| 104 | Ernst | 6000 | 6900 | 900 |

…………

20 rows selected

1. Выведите фамилии служащих (первая буква каждой фамилии должна быть заглавной, а остальные - строчными) и длину каждой фамилии для тех служащих, фамилия которых начинается с символа J, A или М. Присвойте соответствующие заголовки столбцам.

SELECT  initcap(last\_name) "Name",

length(last\_name) "Length"

FROM employees

WHERE last\_name like 'J%'

or last\_name like 'A%'

or last\_name like 'M%'

|  |  |
| --- | --- |
| Name | Lenght |
| Abel | 4 |
| Matos | 5 |
| Mourgos | 7 |

## Лабораторная работа 4

* Составление запросов, требующих использования числовых, символьных функций и функций для работы с датами.
* Использование конкатенации с функциями.
* Составление запросов, нечувствительных к регистру символов, для проверки полезности символьных функций.
* Вычисление продолжительности работы служащего в месяцах и годах.
* Определение даты аттестации служащего.

1. Для каждого служащего выведите фамилию и вычислите количество месяцев со дня найма до настоящего времени, округлённое до ближайшего целого. Назовите столбец MONTH\_WORKED. Результаты отсортируйте по количеству отработанных месяцев. Округлите число до ближайшего целого.

SELECT  last\_name, round(months\_between(current\_date, hire\_date)) "MONTH\_WORKED"

FROM employees

ORDER BY monrh\_worked

|  |  |  |
| --- | --- | --- |
| LAST\_NAME |  | MONTH\_WORKED |
| Zlotkey |  | 13 |
| Mourgos |  | 16 |
| Grant |  | 22 |
| Lorentz |  | 25 |
| Vargas |  | 32 |
| Taylor |  | 36 |
| Matos |  | 36 |
| Fay |  | 43 |

……………..

20 rows selected

1. Получите по каждому служащему отчёт в следующем виде:  
   *<фамилия> зарабатывает <оклад> в месяц, но желает <утроенный оклад>*. Назовите столбец Dream Salaries

SELECT  last\_name||' зарабатывает'||to\_char(salary, '$99,999.00')

||' в месяц, но желает'||to\_char(salary\*3, '$999,999.00') "Dream Salaries"

FROM employees

|  |
| --- |
| Dream Salaries |
| King earns $24,000.00 monthly but wants $72,000.00 |
| Kochhar earns $17,000.00 monthly but wants $51,000.00 |
| De Haan earns $17,000.00 monthly but wants $51,000.00 |
| Hunold earns $9,000.00 monthly but wants $27,000.00 |
| Ernst earns $6,000.00 monthly but wants $18,000.00 |
| Lorentz earns $4,200.00 monthly but wants $12,600.00 |
| Mourgos earns $5,800.00 monthly but wants $17,400.00 |
| Rajs earns $3,500.00 monthly but wants $10,500.00 |
| Davies earns $3,100.00 monthly but wants $9,300.00 |

……………

20 rows selected

1. Напишите запрос для вывода фамилий и окладов всех служащих. Назовите выходной столбец SALARY. Длина столбца SALARY – 15 символов с заполнением символом $.

SELECT  last\_name, lpad(salary, 15, '$') "SALARY"

FROM employees

|  |  |
| --- | --- |
| LAST\_NAME | SALARY |
| King | $$$$$$$$$$24000 |
| Kochhar | $$$$$$$$$$17000 |
| De Haan | $$$$$$$$$$17000 |
| Hunold | $$$$$$$$$$$9000 |
| Ernst | $$$$$$$$$$$6000 |
| Lorentz | $$$$$$$$$$$4200 |
| Mourgos | $$$$$$$$$$$5800 |
| Rajs | $$$$$$$$$$$3500 |
| Davies | $$$$$$$$$$$3100 |
| Matos | $$$$$$$$$$$2600 |

………………

20 rows selected

1. Для каждого служащего выведите фамилию, дату найма и дату пересмотра зарплаты, которая приходится на первый понедельник после 6 месяцев работы. Назовите столбец REVIEW. Формат даты при выводе имеет вид: “Monday, the Thirty-First of MM, YYYY”.

SELECT last\_name, hire\_date,

to\_char(next\_day(add\_month(hire\_date, 6), 'Понедельник'),'fmDay, ddspth "of" Month, yyyy') "REVIEW"

FROM employees

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | HIRE\_DATE | REVIEW |
| King | 17-JUN-87 | Monday, the Thirty-First of December, 1987 |
| Kochhar | 21-SEP-89 | Monday, the Thirty-Sixth of March, 1990 |
| De Haan | 13-JAN-93 | Monday, the Nineteenth of July, 1993 |
| Hunold | 03-JAN-90 | Monday, the Ninth of July, 1990 |

…………………

20 rows selected

1. По каждому служащему выведите фамилию, дату найма и день недели, когда он был нанят на работу. Назовите последний столбец DAY. Отсортируйте результаты по датам.

SELECT last\_name, hire\_date, to\_char(hire\_date, 'Day') "DAY"

FROM employees

ORDER BY hire\_date

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | HIRE\_DATE | DAY |
| Grant | 24-MAY-99 | MONDAY |
| Ernst | 21-MAY-91 | TUESDAY |
| Mourgos | 16-NOV-99 | TUESDAY |
| Taylor | 24-MAR-98 | TUESDAY |
| Rajs | 17-OCT-95 | TUESDAY |
| Gietz | 07-JUN-94 | TUESDAY |
| Higgins | 07-JUN-94 | TUESDAY |
| King | 17-JUN-87 | WEDNESDAY |
| De Haan | 13-JAN-93 | WEDNESDAY |

………………

20 rows selected

1. Напишите запрос для вывода фамилии и суммы комиссионных каждого служащего. Если служащий не зарабатывает комиссионных, укажите в столбце «No Commission». Назовите столбец COMM.

SELECT last\_name,

nvl (to\_char (commission\_pct), 'No Commission') “COMM“

FROM employees

|  |  |
| --- | --- |
| LAST\_NAME | COMM |
| King | No Commission |
| Kochhar | No Commission |
| De Haan | No Commission |
| Hunold | No Commission |
| Ernst | No Commission |
| Lorentz | No Commission |
| Mourgos | No Commission |
| Rajs | No Commission |
| Davies | No Commission |
| Matos | No Commission |
| Vargas | No Commission |
| Zlotkey | .2 |
| Abel | .3 |

…………….

20 rows selected

1. Используя функцию DECODE, напишите запрос для отображения должности сотрудника и её разряда (grade). Разряд каждого типа должности JOB\_ID приведён в таблице.

SELECT  job\_id,

decode(job\_id, 'AD\_PRES', 'A',

'ST\_MAN', 'B',

'IT\_PROG', 'C',

'SA\_REP', 'D',

'ST\_CLERK', 'E',

Job\_id, 0) "G"

FROM employees

|  |  |
| --- | --- |
| Должность | Разряд |
| *AD\_PRES* | A |
| *ST\_MAN* | B |
| *IT\_PROG* | C |
| *SA\_REP* | D |
| *ST\_CLERK* | E |
| Другая | 0 |

|  |  |
| --- | --- |
| JOB\_ID | G |
| AD\_PRES | A |
| AD\_VP | 0 |
| AD\_VP | 0 |
| IT\_PROG | C |
| IT\_PROG | C |
| IT\_PROG | C |
| ST\_MAN | B |
| ST\_CLERK | E |

…………………..

20 rows selected

1. Перепешите команду из предыдущего задания, используя синтаксис выражения CASE.

SELECT  job\_id,

CASE job\_id

WHEN 'AD\_PRES' THEN 'A'

WHEN 'ST\_MAN' THEN 'B'

WHEN 'IT\_PROG' THEN 'C'

WHEN 'SA\_REP' THEN 'D'

WHEN 'ST\_CLERK' THEN 'E'

ELSE '0'

END

AS "G"

FROM employees

# Лабораторная работа 5

* Соединение таблиц с использованием эквисоединения.
* Выполнение внешних соединений и соединений таблицы с собой
* Включение дополнительных условий.

1. Напишите запрос для вывода фамилии, номера отдела и названия отдела дл всех служащих.

SELECT e.last\_name, d.department\_id, d.department\_name

FROM departments d

RIGHT OUTER JOIN employees e

ON d.department\_id = e.department\_id

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | DEPARTMENT\_ID | DEPARTMENT\_NAME |
| Whalen | 10 | Administration |
| Hartstein | 20 | Marketing |
| Fay | 20 | Marketing |
| Mourgos | 50 | Shipping |
| Rajs | 50 | Shipping |
| Davies | 50 | Shipping |
| Matos | 50 | Shipping |
| Vargas | 50 | Shipping |

……………..

19 rows selected

1. Выведите список всех должностей в отдел 80 (должности в списке не должны повторяться) и местоположение отдела.

SELECT distinct job\_id, location\_id

FROM employees JOIN departments

USING (department\_id)

WHERE department\_id = 80

|  |  |
| --- | --- |
| JOB\_ID | LOCATION\_ID |
| SA\_MAN | 2500 |
| SA\_REP | 2500 |

1. Напишите запрос для вывода фамилий, названия отдела, идентификатора местоположения отдела и города, в котором он находится, для всех служащих, зарабатывающих комиссионные.

SELECT last\_name, department\_name, location\_id, city

FROM employees

JOIN departments

USING (department\_id)

JOIN locations

USING (location\_id)

WHERE commission\_pct is not null

|  |  |  |  |
| --- | --- | --- | --- |
| LAST\_NAME | DEPARTMENT\_NAME | LOCATION\_ID | CITY |
| Zlotkey | Sales | 2500 | Oxford |
| Abel | Sales | 2500 | Oxford |
| Taylor | Sales | 2500 | Oxford |

1. Выведите фамилии всех служащих, содержащтих букву «а» (в строчном регистре), с названиями отделов. Сохраните свою команду SQL в файле lab4\_4.sql.

SELECT last\_name, department\_name

FROM employees JOIN departments

USING (department\_id)

WHERE last\_name LIKE '%a%'

|  |  |
| --- | --- |
| LAST\_NAME | DEPARTMENT\_NAME |
| Whalen | Administration |
| Hartstein | Marketing |
| Fay | Marketing |
| Rajs | Shipping |
| Davies | Shipping |
| Matos | Shipping |
| Vargas | Shipping |
| Taylor | Sales |
| Kochhar | Executive |
| De Haan | Executive |

10 rows selected.

1. Напишите запрос для вывода фамилии, должности, номера отдела и названия отдела всех служащих, работающих в городе Toronto.

SELECT last\_name, job\_id, department\_id, department\_name

FROM employees

JOIN departments

USING(department\_id)

JOIN locations

USING(location\_id)

WHERE city = 'Toronto'

|  |  |  |  |
| --- | --- | --- | --- |
| LAST\_NAME | JOB\_ID | DEPARTMENT\_ID | DEPARTMENT\_NAME |
| Hartstein | MK\_MAN | 20 | Marketing |
| Fay | MK\_REP | 20 | Marketing |

1. Выведите фамилии и номера всех служащих вместе с фамилиями и номерами их менеджеров. Назовите столбцы EMPLOYEE, Emp#, Manager и Mgr#. Сохраните свою команду SQL в файле lab4\_6.sql.

SELECT e.last\_name "Employee", e.employee\_id "EMP#", m.last\_name "Manager", m.manager\_id "Mgr#"

FROM employees e

JOIN employees m

ON(e.manager\_id = m.manager\_id)

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | EMP# | Manager | Mgr# |
| Kochhar | 101 | King | 100 |
| De Haan | 102 | King | 100 |
| Mourgos | 124 | King | 100 |
| Zlotkey | 149 | King | 100 |

…………….

1. Измените файл lab4\_6.sql так, чтобы получить фамилии всех служащих, включая Кинга, который не имеет менеджера. Упорядочьте результат по возрастанию номера служащего. Сохраните изменённый запрос в текстовом файле lab4\_7.sql. Выполните запрос.

SELECT e.last\_name AS "Employee", e.employee\_id AS "EMP#", m.last\_name AS "Manager", m.manager\_id AS "Mgr#"

FROM employees e

LEFT JOIN employees m

ON (e.manager\_id = m.manager\_id)

ORDER BY "EMP#"

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | EMP# | Manager | Mgr# |
| King | 100 |  |  |
| Kochhar | 101 | King | 100 |
| De Haan | 102 | King | 100 |
| Hunolds | 103 | De Haan | 102 |
| Ernst | 104 | Hunold | 103 |
| Lorentz | 107 | Hunold | 103 |

…………………………….

1. Создайте запрос для вывода номера отдела, фамилии служащего и фамилий всех служащих, работающих в одном отделе с данным служащим. Дайте столбцам соответствующие имена.

SELECT department\_id "DEPARTMENT",

e.last\_name "EMPLOYEE",

c.last\_name "COLLEAGUE"

FROM employees c

JOIN employees e

USING(department\_id)

|  |  |  |
| --- | --- | --- |
| DEPARTMENT | EMPLOYEE | COLLEAGUE |
| 20 | Fay | Hartstein |
| 20 | Hartstein | Fay |
| 50 | Davies | Matos |
| 50 | Davies | Mourgos |
| 50 | Davies | Rajs |
| 50 | Davies | Vargas |

…………………

1. Покажите структуру таблицы JOB\_GRADES. Создайте запрос для вывода фамилии, должности, названия отдела, оклада и категории (GRADE\_LEVEL) всех служащих.

DESCRIBE job\_grades

SELECT e.last\_name, e.job\_id, d.department\_name, e.salary, g.grade\_level

FROM departments d

JOIN employees e

USING(department\_id)

JOIN job\_grades g

ON e.salary

BETWEEN g.lowest\_sal AND g.highest\_sal

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| GRADE\_LEVEL |  | VARCHAR2(3) |
| LOWEST\_SAL |  | NUMBER |
| HIGHEST\_SAL |  | NUMBER |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LAST\_NAME | JOB\_ID | DEPARTMENT\_NAME | SALARY | GRA |
| Matos | ST\_CLERK | Shipping | 2600 | A |
| Vargas | ST\_CLERK | Shipping | 2500 | A |
| Lorentz | IT\_PROG | IT | 4200 | B |
| Mourgos | ST\_MAN | Shipping | 5800 | B |
| Rajs | ST\_CLERK | Shipping | 3500 | B |
| Davies | ST\_CLERK | Shipping | 3100 | B |
| Whalen | AD\_ASST | Administration | 4400 | B |

1. Создайте запрос для вывода фамилий и дат найма всех служащих, нанятых после Davies.

SELECT af.last\_name, af.hire\_date

FROM employees af JOIN employees b

ON (af.hire\_date > b.hire\_date)

WHERE b.last\_name = 'Davies'

|  |  |
| --- | --- |
| LAST\_NAME | HIRE\_DATE |
| Lorentz | 07-FEB-99 |
| Mourgos | 16-NOV-99 |
| Matos | 15\_MAR-98 |
| Vargas | 09-JUL-98 |
| Zlotkey | 29-JAN-00 |
| Taylor | 24-MAR-98 |
| Grant | 24-MAY-99 |
| Fay | 17-AUG-97 |

1. По всем служащим нанятым раньше своих менеджеров, выведите фамилии и даты найма самих служащих, а также фамилии и даты найма их менеджеров. Назовите столбцы Employee, Emp, Manager и Manager Hired.

SELECT e.last\_name "Employee", e.hire\_date "Emp Hired", m.last\_name "Manager", m.hire\_date "Mgr hired"

FROM employees e JOIN employees m

ON (m.hire\_date > e.hire\_date)

WHERE m.employee\_id = e.manager\_id

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | Emp Hired | Manager | Mgr hired |
| Whalen | 17-SEP-87 | Kochhar | 21-SEP-89 |
| Hunold | 03-JAN-90 | De Haan | 13-JAN-93 |
| Rajs | 17-OCT-95 | Mourgos | 16-NOV-99 |
| Davies | 29-JAN-97 | Mourgos | 16-NOV-99 |
| Matos | 15-MAR-98 | Mourgos | 16-NOV-99 |
| Vargas | 09-JUL-98 | Mourgos | 16-NOV-99 |
| Abel | 11-MAY-96 | Zlotkey | 29-JAN-00 |
| Taylor | 24-MAR-98 | Zlotkey | 29-JAN-00 |
| Grant | 24-MAY-99 | Zlotkey | 29-JAN-00 |

1. Выведите номера, наименования и местоположение всех отделов, а также количество работающих в них сотрудников. Обеспечьте вывод отделов в которых нет сотрудников.

SELECT d. department\_id, d. department\_name, d. location\_id, count (e. employee\_id)

FROM departments d

LEFT JOIN employees e

ON d.department\_id = e.department\_id

GROUP BY d. department\_id, d. department\_name, d. location\_id

|  |  |  |  |
| --- | --- | --- | --- |
| DEPARTMENT\_ID | DEPARTMENT\_NAME | LOCATION\_ID | COUNT(E.EMPLOYEE\_ID) |
| 10 | Administration | 1700 | 1 |
| 20 | Marketing | 1800 | 2 |
| 50 | Shipping | 1500 | 5 |
| 60 | IT | 1400 | 3 |
| 80 | Sales | 2500 | 3 |
| 90 | Executive | 1700 | 3 |
| 110 | Accounting | 1700 | 2 |
| 190 | Contracting | 1700 | 0 |

1. Какие должности имеются в отделах Administration и Executive, а также сколько сотрудников занимают эти должности? Выведите первыми наиболее часто встречаемые должности.

SELECT e.job\_id, count(e.employee\_id) "FREQUENCY"

FROM employees e RIGHT OUTER JOIN departments d

ON (d.department\_id = e.department\_id)

WHERE department\_name = 'Administration' OR department\_name = 'Executive'

GROUP BY e.job\_id

ORDER BY FREQUENCY DESC

|  |  |
| --- | --- |
| JOB\_ID | FREQUENCY |
| AD\_VP | 2 |
| AD\_ASST | 1 |
| AD\_PRES | 1 |

1. Покажите сотрудников, менеджеры которых имеют оклад выше $15000. выведите следующие сведения: фамилию сотрудника, фамилию менеджера, его оклад и уровень оклада (GRADE\_LEVEL).

SELECT e.last\_name, m.last\_name MANAGER, m.salary, j.grade\_level GRA

FROM employees m JOIN job\_grades j

ON (m.salary > 15000)

JOIN employees e

ON (m.employee\_id = e.manager\_id)

|  |  |  |  |
| --- | --- | --- | --- |
| LAST\_NAME | MANAGER | SALARY | GRA |
| Kochhar | King | 24000 | E |
| De Haan | King | 24000 | E |
| Mourgos | King | 24000 | E |
| Zlotkey | King | 24000 | E |
| Hartstein | King | 24000 | E |
| Whalen | Kochhar | 17000 | E |
| Higgins | Kochhar | 17000 | E |
| Hunold | De Haan | 17000 | E |

## Лабораторная работа 6

* Использование групповых функций
* Вывод данных по группам с помощью предложения GROUP BY
* Включение и исключение групп с помощью предложения HAVING

1. Напишите запрос для вывода самого высокого, самого низкого и среднего оклада по всем служащим, а также суммы всех окладов. Назовите столбцы Maximum, Minimum, Average и Sum. Округлите суммы до ближайшего целого значения. Сохраните свой запрос в файле lab5\_6.sql.

SELECT

max (salary) "Maximum",

min (salary) "Minimum",

sum (salary) "Sum",

round (avg (salary), 0) "Average"

FROM employees

|  |  |  |  |
| --- | --- | --- | --- |
| Maximum | Minimum | Sum | Average |
| 24000 | 2500 | 175500 | 8775 |

1. Измените запрос в lab5\_6.sql. так, чтобы получить самый низкий, самый высокий и средний оклады, а также сумму окладов по каждой должности. Сохраните изменённый запрос в файле lab5\_5.sql. Выполните запрос из lab5\_5.sql.

SELECT job\_id,

max (salary) "Maximum",

min (salary) "Minimum",

sum (salary) "Sum",

round (avg (salary), 0) "Average"

FROM employees

GROUP BY job\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JOB\_ID | MAXIMUM | MINIMUM | SUM | AVERAGE |
| AC\_ACCOUNT | 8300 | 8300 | 8300 | 8300 |
| AC\_MGR | 12000 | 12000 | 12000 | 12000 |
| AD\_ASST | 4400 | 4400 | 4400 | 4400 |
| AD\_PRES | 24000 | 24000 | 24000 | 24000 |
| AD\_VP | 17000 | 17000 | 34000 | 17000 |

…………….

1. Напишите запрос для вывода должности и количества служащих, занимающих каждую должность.

SELECT job\_id, count (\*)

FROM employees

GROUP BY job\_id

|  |  |
| --- | --- |
| JOB\_ID | COUNT(\*) |
| AD\_ACCOUNT | 1 |
| AC\_MGR | 1 |
| AD\_ASST | 1 |
| AD\_PRES | 1 |
| AD\_VP | 2 |
| IT\_PROG | 3 |
| MK\_MAN | 1 |
| MK\_REP | 1 |
| SA\_MAN | 1 |
| SA\_REP | 3 |
| ST\_CLERK | 4 |
| ST\_MAN | 1 |

1. Получите количество служащих, имеющих подчинённых, без их перечисления. Назовите столбец Number of Managers. Используйте столбец MANAGER\_ID для определения числа менеджеров.

SELECT count (distinct manager\_id) "Number of Managers"

FROM employees

|  |
| --- |
| Number of Managers |
| 8 |

1. Напишите запрос для вывода разности между самым высоким и самым низким окладами. Назовите столбец DIFFERENCE

SELECT (max (salary) - min (salary)) "DIFFERENCE"

FROM employees

|  |
| --- |
| DIFFERENCE |
| 21500 |

1. Напишите запрос для вывода номера каждого менеджера, имеющего подчинённых, и заработную плату самого низкооплачиваемого из его подчинённых. Исключите менеджеров для которых неизвестны их менеджеры. Исключите все группы, где минимальный оклад составляет менее $6000. Отсортируйте выходные строки в порядке убывания оклада.

SELECT manager\_id, min (salary) "MIN(SALARY)"

FROM employees

WHERE manager\_id is not null

HAVING min (salary) >= 6000

GROUP BY manager\_id

ORDER BY MIN(SALARY) DESC

|  |  |
| --- | --- |
| MANAGER\_ID | MIN(SALARY) |
| 102 | 9000 |
| 205 | 8300 |
| 149 | 7000 |

1. Напишите запрос для вывода названия отдела, местоположения отдела, количества служащих и среднего оклада по этому отделу. Назовите столбцы Name, Location, Number of People и Salary. Округлите средний оклад до ближайшего целого значения.

SELECT

d.department\_name "Name",

d.location\_id "Location",

count (employee\_id) "Number of People",

round (avg (salary), 0) "Salary"

FROM departments d

JOIN employees e

USING (department\_id)

GROUP BY d.department\_name, d.location\_id

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Location | Number of People | Salary |
| Accounting | 1700 | 2 | 10150 |
| Administration | 1700 | 1 | 4400 |
| Executive | 1700 | 3 | 19333.33 |
| IT | 1400 | 3 | 6400 |
| Marketing | 1800 | 2 | 9500 |
| Sales | 2500 | 3 | 10033.33 |
| Shipping | 1500 | 5 | 3500 |

1. Напишите запрос для вывода общего количества служащих и количества служащих, нанятых в 1995, 1996, 1997 и 1998 годах. Дайте соответствующие заголовки столбцам.

SELECT count (\*) "TOTAL",

sum (decode (to\_char (hire\_date, 'YYYY'), 95, 1)) "1995",

sum (decode (to\_char (hire\_date, 'YY'), 96, 1)) "1996",

sum (decode (to\_char (hire\_date, 'YY'), 97, 1)) "1997",

sum (decode (to\_char (hire\_date, 'YY'), 98, 1)) "1998"

FROM employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TOTAL | 1995 | 1996 | 1997 | 1998 |
| 20 | 1 | 2 | 2 | 3 |

1. Напишите матричный запрос для вывода всех должностей и суммы заработной платы служащих, работающих в этой должности в отделах 20, 50, 80 и 90. Последний столбец должен содержать сумму заработной платы служащих этих отделов, занимающих каждую конкретную должность. Дайте столбцам соответствующие заголовки.

SELECT job\_id "Job",

sum (decode (department\_id, 20, salary)) "Dept20",

sum (decode (department\_id, 50, salary)) "Dept50",

sum (decode (department\_id, 80, salary)) "Dept80",

sum (decode (department\_id, 90, salary)) "Dept90",

SUM (salary) "Total"

FROM employees

GROUP BY job\_id

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job | Dept20 | Dept50 | Dept80 | Dept90 | Total |
| AC\_ACCOUNT |  |  |  |  | 8300 |
| AC\_MGR |  |  |  |  | 12000 |
| AD\_ASST |  |  |  |  | 4400 |
| AD\_PRES |  |  |  | 24000 | 24000 |
| AD\_VP |  |  |  | 34000 | 34000 |
| IT\_PROG |  |  |  |  | 19200 |

……………………………

## Лабораторная работа 7

* + Создание подзапросов для выборки данных по неизвестным критериям
  + Использование подзапросов для выявления значений, существующих в одном наборе данных и отсутствующих в другом

1. Создайте запрос для вывода фамилии и даты найма каждого служащего, работающего в одном отделе с Zlotkey. Исключите Zlotkey из выходных данных.

SELECT last\_name, hire\_date

FROM employees

WHERE department\_id = (SELECT department\_id

FROM employees

WHERE last\_name = 'Zlotkey')

AND last\_name <> 'Zlotkey'

1. Создайте запрос для вывода номеров и фамилий всех служащих, оклад которых выше среднего. Отсортируйте данные в порядке увеличения окладов.

SELECT employee\_id, last\_name, salary

FROM employees

WHERE salary > (SELECT avgG(salary)

FROM employees)

ORDER BY salary ASC

1. Создайте запрос для вывода номеров и фамилий всех служащих, работающих в одном отделе с любым служащим, фамилия которого содержит букву «u». Сохраните свой запрос в тестовом файле lab6\_3.sql. Выполните запрос.

SELECT employee\_id, last\_name

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM employees

WHERE last\_name LIKE '%u%')

1. Создайте запрос для вывода фамилии, номера отдела и должности каждого служащего, идентификатор местоположения отдела которого равен 1700.

SELECT last\_name, department\_id, job\_id

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM departments

WHERE location\_id = 1700)

1. Получите список фамилий и окладов всех служащих, подчинённых Кингу.

SELECT last\_name, salary

FROM employees

WHERE manager\_id IN (SELECT employee\_id

FROM employees

WHERE last\_name = 'King')



1. Получите номер отдела, фамилию и должность для каждого служащего, работающего в администрации (department\_name = ‘Executive’).

SELECT department\_id, last\_name, job\_id

FROM employees

WHERE department\_id = (SELECT department\_id

FROM departments

WHERE department\_name = 'Executive')

1. Измените lab6\_3.sql для вывода номеров, фамилий и окладов всех служащих, которые работают в одном отделе с любым служащим с буквой «u» в фамилии и оклады которых превышают средний. Сохраните новый запрос в файле lab6\_7.sql. Выполните его.

SELECT employee\_id, last\_name, salary

FROM employees

WHERE department\_id IN (SELECT department\_id

FROM employees

WHERE last\_name LIKE '%u%')

AND salary>(SELECT avg(salary)

FROM employees)



1. Покажите номер отдела с наивысшей средней заработной платой и наименьший оклад работающего в нём сотрудника.

SELECT department\_id, min(salary)

FROM employees

GROUP BY department\_id

HAVING avg(salary) = (SELECT max(avg(salary))

FROM employees

GROUP BY department\_id)

1. Выведите номера , наименования и местоположения отделов, в которых не работают торговые представители (job\_id = ‘SA\_REP’).

SELECT department\_id, department\_name, manager\_id, location\_id

FROM departments

WHERE department\_id != ALL

(SELECT NVL(department\_id, 0)

FROM employees

WHERE job\_id = 'SA\_REP')

## Лабораторная работа 8

* + Создание запросов, в которых используются операторы SET.
  + Альтернативные методы соединения.

1. Используя оператор SET, выведите номера отделов (department\_id), в которых нет служащих с идентификатором должности (job\_id) ST\_CLERK.



SELECT department\_id

FROM departments

MINUS

SELECT department\_id

FROM employees

WHERE job\_id LIKE 'ST\_CLERK'

1. Используя оператор SET, выведите идентификаторы и наименования стран, в которых не располагаются отделы компании. 

SELECT country\_id, country\_name

FROM countries

WHERE country\_id IN (SELECT country\_id

FROM countries

MINUS

SELECT country\_id

FROM locations

WHERE location\_id IN (SELECT location\_id FROM departments))

1. Используя оператор SET, выведите список должностей отделов 10, 50 и 20 в таком же порядке отделов. Выведите столбцы job\_id и department\_id. 

SELECT job\_id, department\_id

FROM (SELECT job\_id, department\_id

FROM employees

WHERE department\_id IN (10, 20, 50)

INTERSECT

SELECT job\_id, department\_id

FROM employees)

ORDER BY decode (department\_id, 10, 1, 20, 3, 50, 2)

1. Выведите номер сотрудника и идентификатор его должности, если его текущая должность совпадает с той, которую он уже занимал, работая в компании. 

SELECT employee\_id, job\_id

FROM employees

INTERSECT

SELECT employee\_id, job\_id

FROM job\_history

1. Напишите составной запрос, который выводит следующее:
   * Номера и отделы всех сотрудников из таблицы EMPLOYEES, независимо от того, относятся ли они к какому-то отделу или нет;
   * Номера и наименования всех отделов из таблицы DEPARTMENTS, независимо от того, есть ли в них сотрудники или нет.

SELECT employee\_id, department\_id, to\_char (NULL) department\_name

FROM employees

UNION

SELECT TO\_NUMBER (NULL), department\_id, department\_name

FROM departments

# Лабораторная работа 9

* + Создание многостолбцовых подзапросов
  + Написание связанных подзапросов
  + Использование оператора EXIST
  + Использование скалярных подзапросов

1. Создайте запрос для вывода фамилии, номера отдела и оклада всех служащих, чей номер отдела и оклад совпадают с номером отдела и окладом любого служащего, зарабатывающего комиссионные. 

SELECT last\_name, department\_id, salary

FROM employees

WHERE (department\_id, salary) IN

(SELECT department\_id, salary

FROM employees

WHERE commission\_pct IS NOT NULL)

1. Выведите фамилию, название отдела и оклад всех служащих, чей оклад и комиссионные совпадают с окладом и комиссионными любого служащего, работающего в отделе, идентификатор местоположения которого Location\_ID = 1700. 

SELECT e.last\_name, d.department\_name, e.salary

FROM employees e, departments d

WHERE e.department\_id = d.department\_id

AND (salary, NVL(commission\_pct, 0)) IN

(SELECT salary, NVL(commission\_pct,0)

FROM employees e, departments d

WHERE e.department\_id = d.department\_id

AND d.location\_id = 1700)

1. Создайте запрос для вывода фамилии, даты найма и оклада всех служащих, которые получают такой же оклад и такие же комиссионные, как Kochhar. Не выводите данные о сотруднике Kochhar.

SELECT a.last\_name, a.hire\_date, a.salary

FROM employees a

WHERE(a.hire\_date, a.salary) IN

(SELECT b.hire\_date, b.salary

FROM employees b

WHERE EXISTS

(SELECT c.last\_name

FROM employees c

WHERE last\_name = 'Kochhar'

AND b.salary = c.salary)

AND last\_name <> 'Kochhar')

1. Выведите фамилию, должность и оклад всех служащих, оклад которых превышает оклад каждого клерка торгового менеджера (JOB\_ID = ‘SA\_MAN’). Отсортируйте результаты по убыванию окладов. 

SELECT last\_name, job\_id, salary

FROM employees

WHERE salary >

(SELECT min(salary)

FROM employees

Where job\_id = 'SA\_MAN' )

AND job\_id <> 'SA\_MAN'

ORDER BY salary DESC

1. Выведите номера, фамилии и отделы служащих, живущих в городах, названия которых начинаются с буквы Т. 

SELECT employee\_id, last\_name, department\_id

FROM employees

WHERE EXISTS

(SELECT city

FROM locations

WHERE city LIKE 'T%')

1. Напишите запрос для нахождения всех сотрудников, которые зарабатывают больше среднего оклада по их отделу. Выведите фамилию, оклад, номер отдела и средний оклад по отделу. Отсортируйте результаты по средней зарплате. Используйте псевдонимы для выбираемых столбцов.

SELECT a.last\_name ENAME, a.salary "SALARY", a.department\_id DEPTNO,

(SELECT AVG(b.salary) FROM employees b

WHERE b.department\_id = a.department\_id)DEPT\_AVG

FROM employees a

WHERE a.salary >

(SELECT AVG(b.salary)

FROM employees b

WHERE b.department\_id = a.department\_id )

ORDER BY DEPT\_AVG

1. Найдите всех сотрудников, не являющихся руководителями. Выполните это с помощью оператора NOT EXIST. 

SELECT a.last\_name

FROM employees a

WHERE NOT EXISTS

(SELECT b.last\_name

FROM employees b

WHERE b.manager\_id = a.employee\_id)

1. Может ли это же быть сделано с помощью оператора NOT IN?

Нет, не будут обработаны значения null

1. Выведите фамилии сотрудников, зарабатывающих меньше среднего оклада по их отделу. 

SELECT e.last\_name

FROM employees e

WHERE salary IN

(SELECT salary

FROM employees

where salary <

(SELECT avg(a.salary)

FROM employees a

WHERE a.department\_id = e.department\_id ))

1. Выведите фамилии сотрудников, у которых есть коллеги по отделу, которые были приняты на работу позже, но имеют более высокий оклад. 

SELECT e.last\_name

FROM employees e

WHERE EXISTS

(SELECT a.last\_name

FROM employees a

WHERE a.hire\_date > e.hire\_date

AND a.salary > e.salary

AND a.department\_id=e.department\_id )

1. Выведите номера, фамилии и наименования отделов всех сотрудников. Используйте скалярный подзапрос в команде SELECT для вывода наименований отделов. 

SELECT e.employee\_id, e.last\_name,

(SELECT d.department\_name

FROM departments d

WHERE d.department\_id = e.department\_id) "DEPARTMENT"

FROM employees e

## Лабораторная работа 10

* + Отличие иерархических запросов от неиерархических
  + Обход дерева
  + Получение отчёта с отступом с помощью псевдостолбца LEVEL
  + Отсечение структур дерева
  + Сортировка выходных данных

1. Напишите отчёт, в котором отражена структура отдела, которым руководит Mourgos. Выведите фамилии, оклады и номер отдела сотрудников. 

SELECT last\_name, salary, department\_id

FROM employees

START WITH last\_name = ('Mourgos')

CONNECT BY PRIOR employee\_id = manager\_id

1. Создайте отчёт, который показывает иерархию менеджеров, которым подчиняется сотрудник Lorentz. Выведите сначала менеджера, перед которым непосредственно отчитывается Lorentz. 

SELECT last\_name

FROM employees

WHERE last\_name != 'Lorentz'

START WITH last\_name = 'Lorentz'

CONNECT BY PRIOR manager\_id = employee\_id

1. Создайте отчёт с отступом, в котором отражается иерархия управления, начиная с сотрудника по фамилии Kochhar. Выведите фамилии, номера менеджеров и номера отделов сотрудников. Назовите столбцы как показано в примере выходных результатов. 

SELECT LPAD(last\_name, lengrh(last\_name)+(LEVEL)-1,'\_') "NAME",

manager\_id "MGR",

department\_id "DEPNO"

FROM employees

START WITH last\_name = 'Kochhar'

CONNECT BY PRIOR employee\_id = manager\_id

1. Создайте отчёт, отражающий иерархию управления компанией. Начните с сотрудника самого высокого уровня и исключите из выходных данных всех служащих с идентификатором должности IT\_PROG, а также сотрудника De Haan и всех, кто перед ним отчитывается. 

SELECT last\_name, employee\_id, manager\_id

FROM employees

WHERE job\_ID != 'IT\_PROG'

AND last\_name != 'De Haan'

AND employee\_id

NOT IN

(SELECT employee\_id

FROM employees

WHERE manager\_id =

(SELECT manager\_id

FROM employees

WHERE last\_name = 'De Haan') )

START WITH manager\_id IS NULL

CONNECT BY PRIOR employee\_id = manager\_id