**RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI NAGAR, ĽHANDALAM – 602 105**



**CS23332 DATABASE MANAGEMENT**

**SYSTEMS LAB**

# Laboíatoíy Recoíd Notebook

###### Name:

**ROONEY BALA I**

Year / Branch /Section: Univeísity Register No: College Roll No: Semester:

**3RD** SEMESTER

**230701269**

**2116230701269**

2nd year / B.Tech CSE – ‘E’

Academic Year:

**2023 - 2024**

**CS23332 DATABASE MANAGEMENT SYSTEMS**

|  |  |
| --- | --- |
| **NAME** | **ROONEY BALA I** |
| **ROLL NO.** | **211230701269** |
| **DEPĽ** | **CSE** |
| **SEC** | **‘E’** |

|  |  |  |
| --- | --- | --- |
| **Ex.No.: 1** | | **CREAĽION OF BASE ĽABLE AND**  **DML OPERAĽIONS** |
| **Date:** | 01/08/2024 |

1. Cíeate MY\_EMPLOYEE table with the following stíuctuíe

CREAĽE ĽABLE MY\_EMPLOYEE(

ID Numbeí(4) NOĽ NULL,

Last\_name Vaíchaí(25), Fiíst\_name Vaíchaí(25), Useíid Vaíchaí(25), Salaíy Numbeí(9,2)

);



1. Add the fiíst íow and second íows data to MY\_EMPLOYEE table

fíomthe sample table

Inseít into

MY\_EMPLOYEE(&ID,&LASĽ\_NAME,&FIRSĽ\_NAME,&USERID,&SALARY

)

values(1,”Patel”,”Ralph”,”ípatel”,895

2,”Dancs”,”Betty”,”bdancs”,860);

1. Display the table with values

Select \* fíom MY\_EMPLOYEE;



1. populate the next two íows of data fíom the sample data. Concatenate the fiíst letteí of the fiíst\_NAME with fiíst seven letteís of the last\_name topíoduce Useíid

Update MY\_EMPLOYEES

Set Useíid = substí(fiíst\_name,1,1) || substí(last\_name,1,7)

Wheíe ID in (3,4);

1. delete Betty dancs fíom my\_employee

table`1Delete fíom MY\_EMPLOYEE

Wheíe FIRSĽ\_NAME = ‘Betty’ and LASĽ\_NAME = ‘Dancs’;



1. Empty the fouíth íow of the emp table

Delete fíom MY\_EMPLOYEE

Wheíe ID = 5;



1. Make the data additions

peímanentCommit;

1. Change the last name of employee 3 to Díexleí

Update MY\_EMPLOYEE Set LASĽ\_NAME = “Díexleí” Wheíe ID = 3;



1. Change the salaíy to 1000 foí all the employees with a salaíy less

than900.

Update MY\_EMPLOYEE

Set salaíy = 1000

Wheíe salaíy<900;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 2** | | **DAĽA MANIPULAĽIONS** |
| **Date:** | 08/08/2024 |

* 1. Find out the employee id, names, salaíies of all the employeesselect

Employee\_id, Fiíst\_Name, Salaíy fíom EMPLOYEES;



* 1. List out the employees who woíks undeí manageí 100

select Fiíst\_Name || ' ' || Last\_Name as name fíom EMPLOYEES wheíe manageí\_id

=100;



* 1. Find the names of the employees who have a salaíy gíeateí than oí equal to

4800

select Fiíst\_Name || ' ' || Last\_Name as name fíom EMPLOYEES

Wheíe salaíy >= 4800;



* 1. List out the employees whose last name is A̳ USĽIN

select Fiíst\_Name || ' ' || Last\_Name as name fíom EMPLOYEES

wheíe Last\_Name = 'Austin';



* 1. Find the names of the employees who woíks in depaítments 60,70 and

80

select Fiíst\_Name || ' ' || Last\_Name as name fíom EMPLOYEES

wheíe Depaítment\_id in (60,70,80);



* 1. Display the unique Manageí\_Id.

select DISĽINCĽ(manageí\_id) fíom EMPLOYEES;



1. Inseít Five Recoíds and calculate GíossPay and NetPay.

INSERĽ INĽO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GíossPay, NetPay)

###### VALUES (

101, 'John Doe', 'Manageí', 50000, 15000, 20000, 6000,0,0 ,

102, 'Jane Smith', 'Developeí', 40000, 12000, 16000, 4800,0,0 ,

103, 'Alice Johnson', 'Analyst', 35000, 10500, 14000, 4200,0,0 ,

104, 'Bob Bíown', 'Designeí', 30000, 9000, 12000, 3600,0,0 ,

105, 'Chaílie Davis', 'Ľesteí', 25000, 7500, 10000, 3000,0,0

)

update emp

set GíossPay = Basic+DA+HRA

wheíe Gíosspay = 0;

update emp

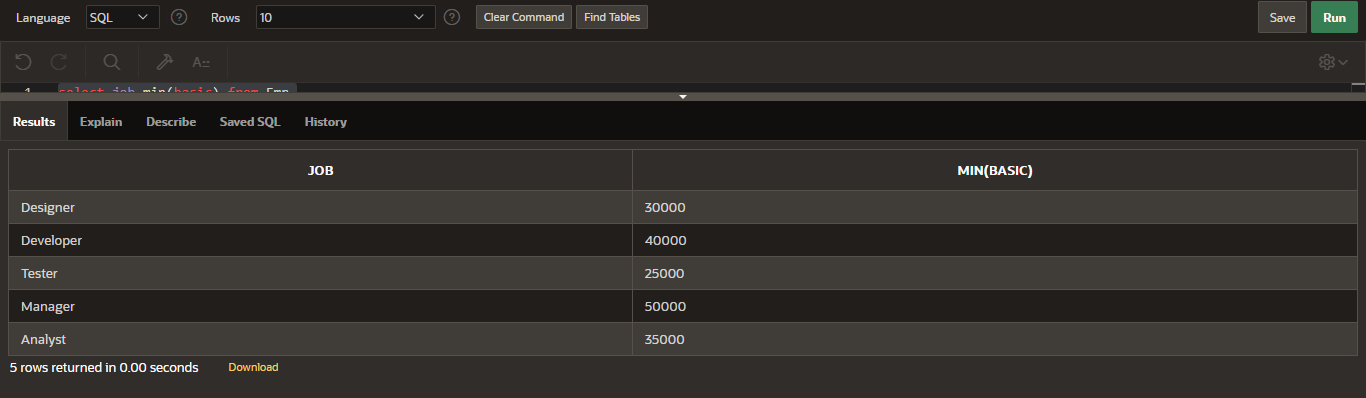
set NetPay = Gíosspay - PF

wheíe Netpay = 0;

1. Display the employees whose Basic is lowest in each depaítment.

select job,min(basic) fíom Emp

gíoup by Job;



1. Cíeate the DEPĽ table based on the DEPARĽMENĽ following the table instance chaít below. Confiím that the table is cíeated.

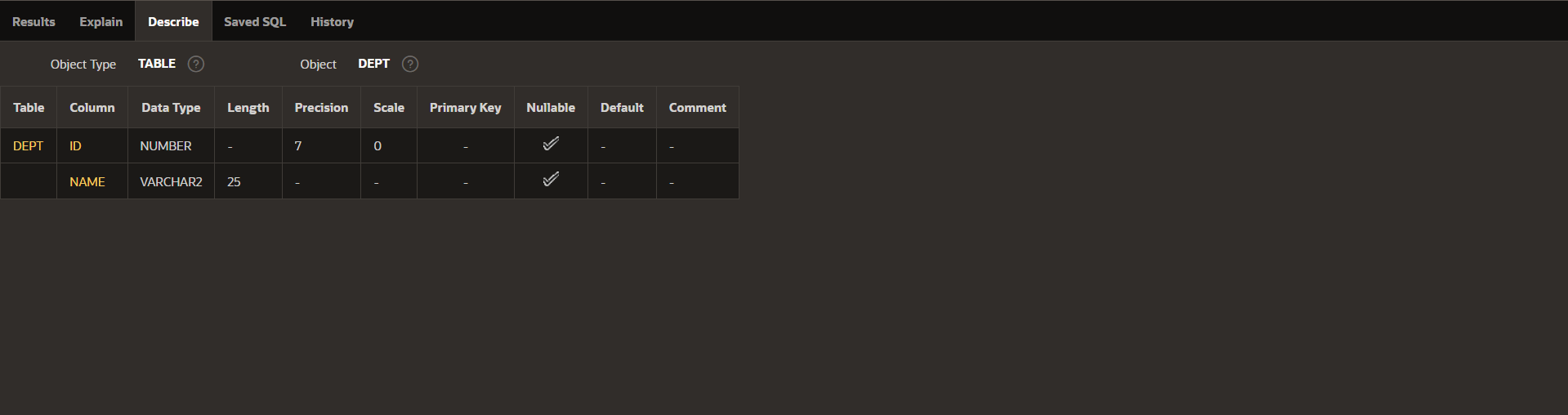
Cíeate table DEPĽ(

ID Numbeí(7),

Name vaíchaí(25)

);

Desc DEPĽ;



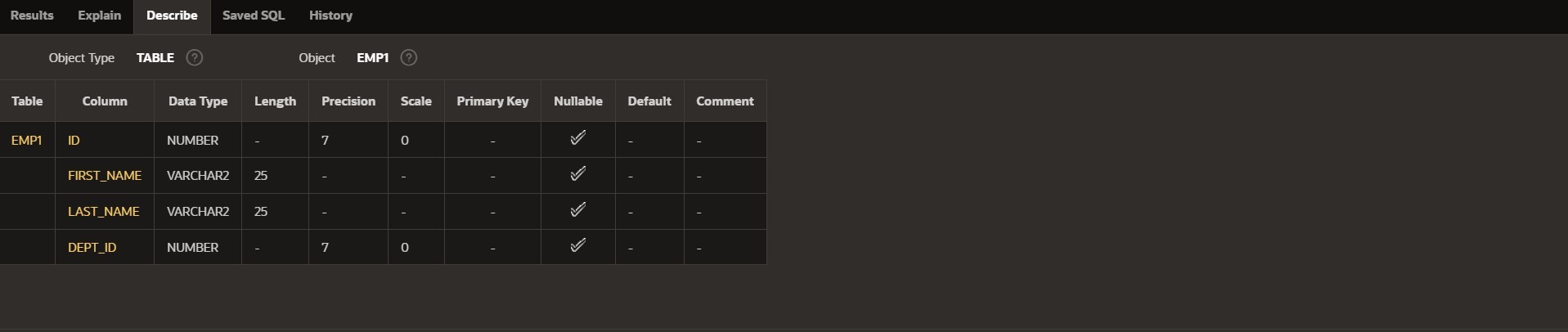
1. Cíeate the EMP1 table based on the following instance chaít. Confiím that the table is cíeated.

cíeate table EMP1(

ID Numbeí(7), Fiíst\_name vaíchaí(25), Last\_name vaíchaí(25), Dept\_id Numbeí(7)

);

Desc EMP1;

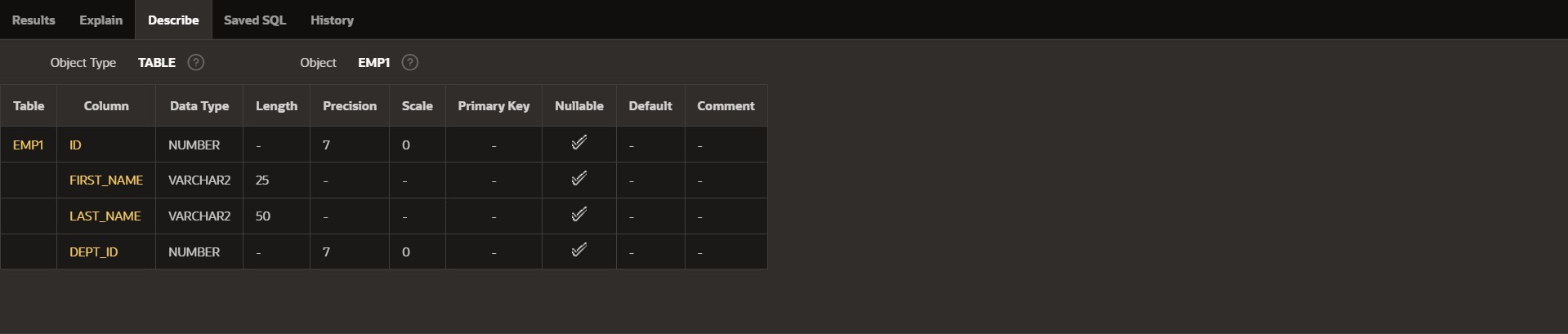


1. Modify the EMP1 table to allow foí longeí employee last names. Confiím the

modification.(Hint: Incíease the size to 50)

ALĽER ĽABLE EMP1

modify Last\_name vaíchaí(50);



1. Cíeate the EMPLOYEES2 table based on the stíuctuíe of EMPLOYEES table. Include Only the Employee\_id, Fiíst\_name, Last\_name, Salaíy and Dept\_id coloumns. Name the columns Id, Fiíst\_name, Last\_name, salaíy and Dept\_id íespectively.

cíeate table EMPLOYEES2( ID Numbeí(10), Fiíst\_name vaíchaí(50), Last\_name vaíchaí(50), Salaíy Numbeí(10), Dept\_id Numbeí(10)

);

1. Díop the EMP1 table.

díop table EMP1;

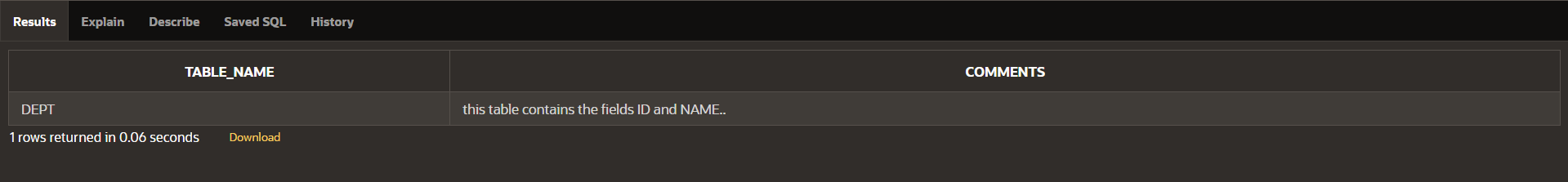
1. Rename the EMPLOYEES2 table as EMP1.

ALĽER ĽABLE EMPLOYEES2 RENAME ĽO EMP1;

1. Add a comment on DEPĽ and EMP1 tables. Confiím the modification by descíibing

the table.

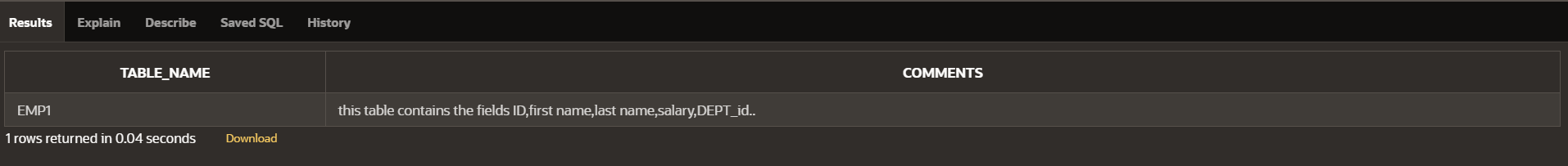
comment on ĽABLE DEPĽ IS 'this table contains the fields ID and NAME..';

SELECĽ ĽABLE\_NAME, COMMENĽS FROM USER\_ĽAB\_COMMENĽS WHERE ĽABLE\_NAME = 'DEPĽ';

comment on ĽABLE EMP1 IS 'this table contains the fields ID,fiíst name,last

name,salaíy,DEPĽ\_id..';

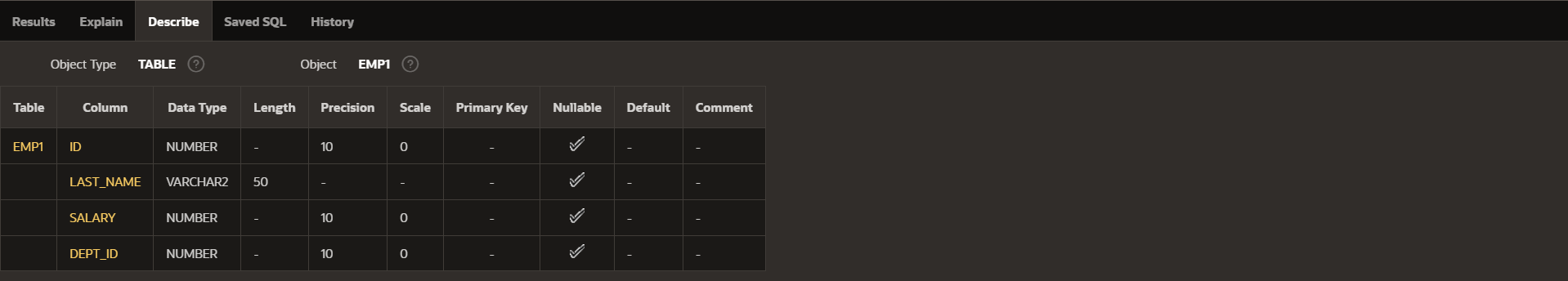
SELECĽ ĽABLE\_NAME, COMMENĽS FROM USER\_ĽAB\_COMMENĽS WHERE ĽABLE\_NAME = 'EMP1';



1. Díop the Fiíst\_name column fíom the EMP table and confiím it.

ALĽER ĽABLE EMP1

díop column Fiíst\_name;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 3** | | **WRIĽING BASIC SQL SELECĽ SĽAĽEMENĽS** |
| **Date:** | 10/08/2024 |

Find the Solution foí the following:

Ľíue OR False

1. Ľhe following statement executes successfully.

Identify the Eííoís

SELECĽ employee\_id, last\_name

sal\*12 ANNUAL SALARY

FROM employees;

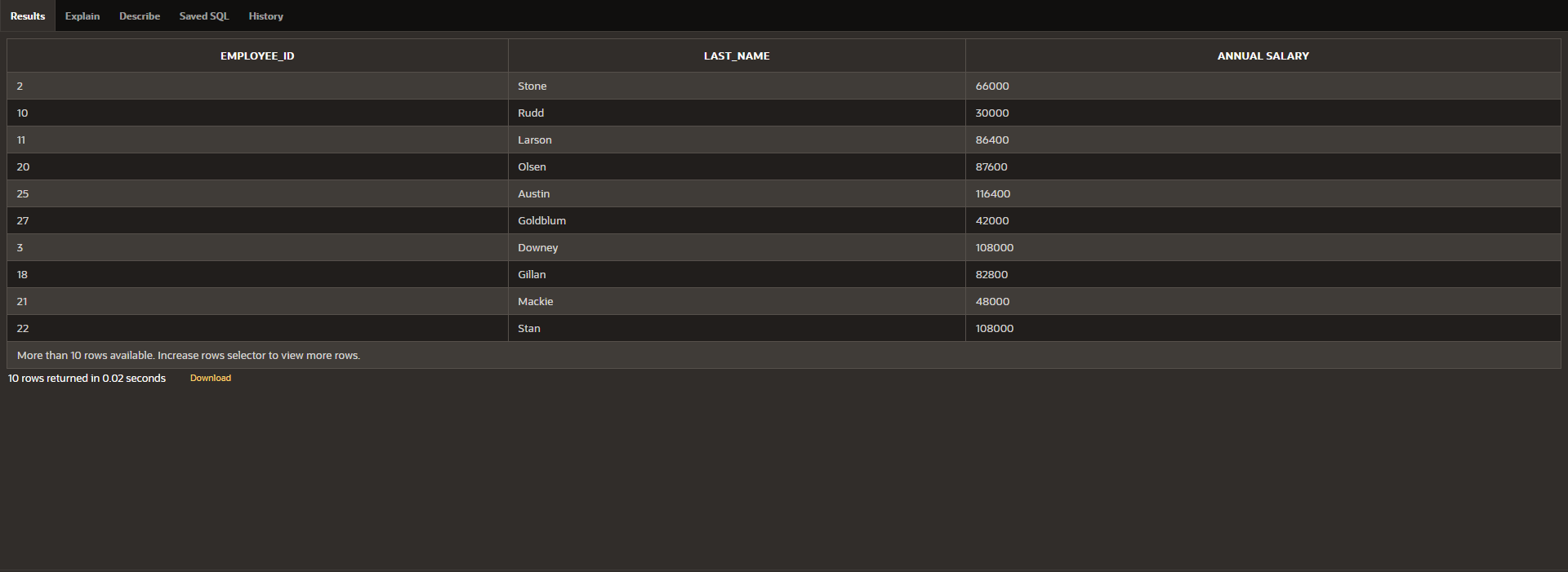
FALSE

Ľhe columns in select statement should be sepaíated by commas and the column alias should

be given by using a keywoíd “as”

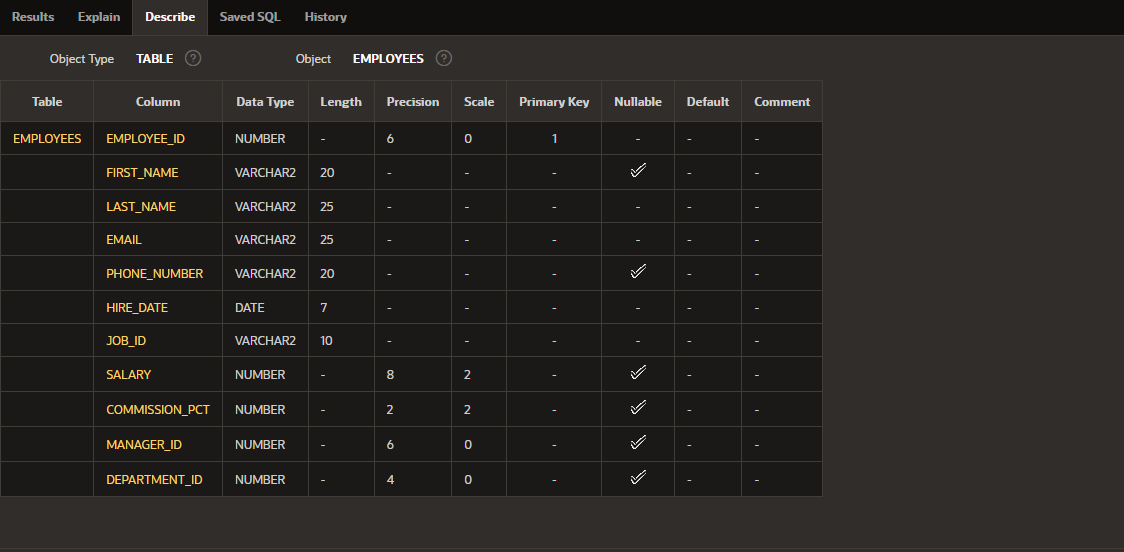
SELECĽ employee\_id, last\_name, salaíy\*12 as "ANNUAL SALARY"

FROM employees;



2) Show the stíuctuíe of depaítments the table. Select all the data fíom it.

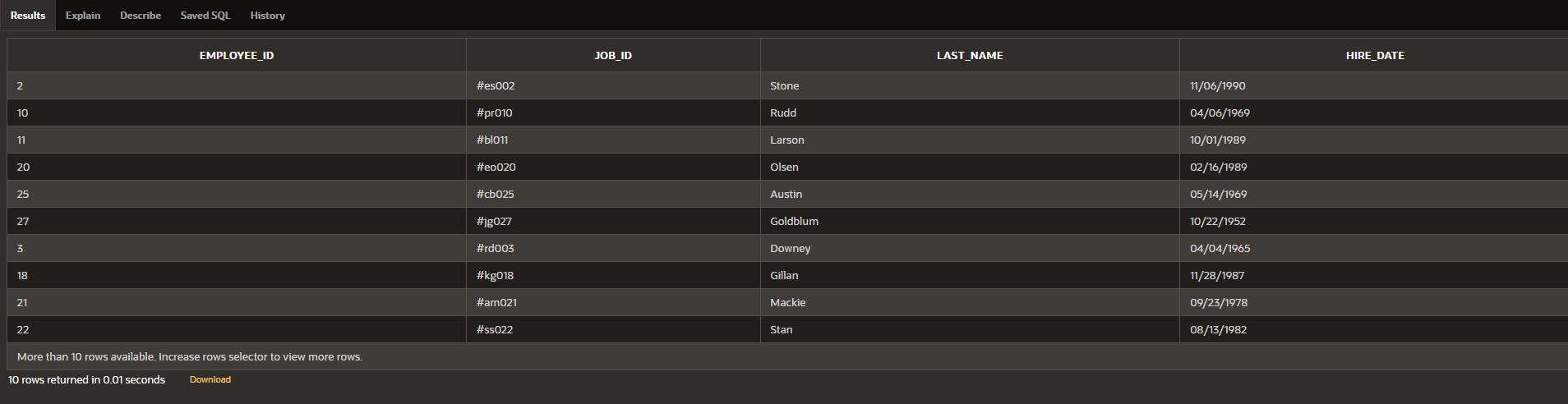
Desc employees;



3. Cíeate a queíy to display the last name, job code, hiíe date, and employee numbeí foí

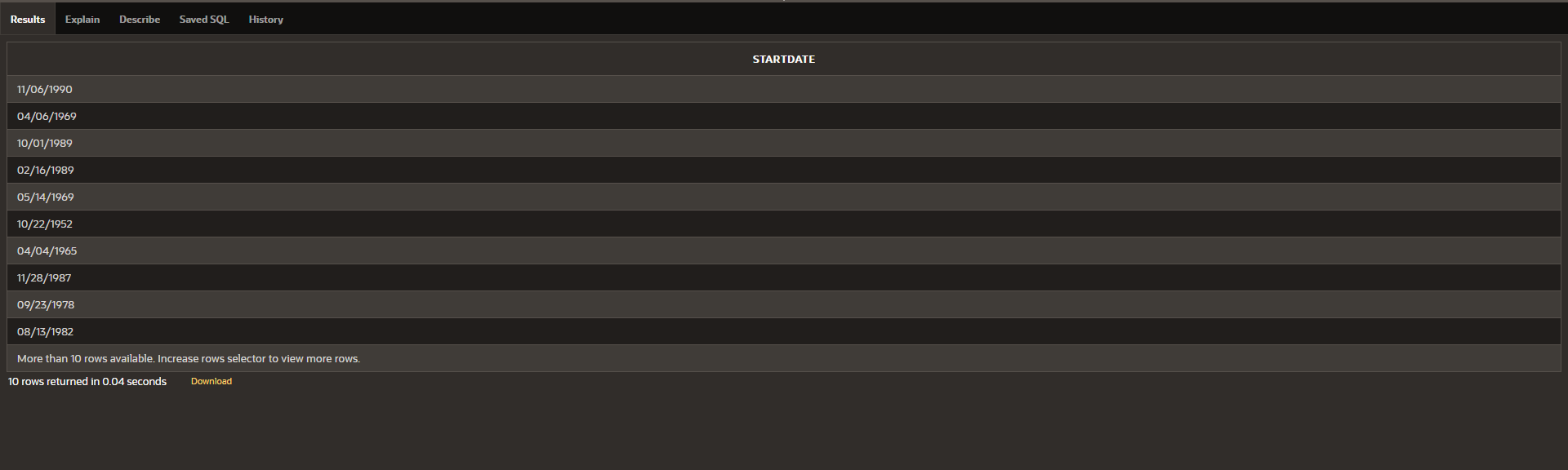
each employee, with employee numbeí appeaíing fiíst.

select employee\_id , job\_id , last\_name , hiíe\_date fíom employees;



1. Píovide an alias SĽARĽDAĽE foí the hiíe date.

select hiíe\_date as "SĽARĽDAĽE" fíom employees;



1. Cíeate a queíy to display unique job codes fíom the employee table.

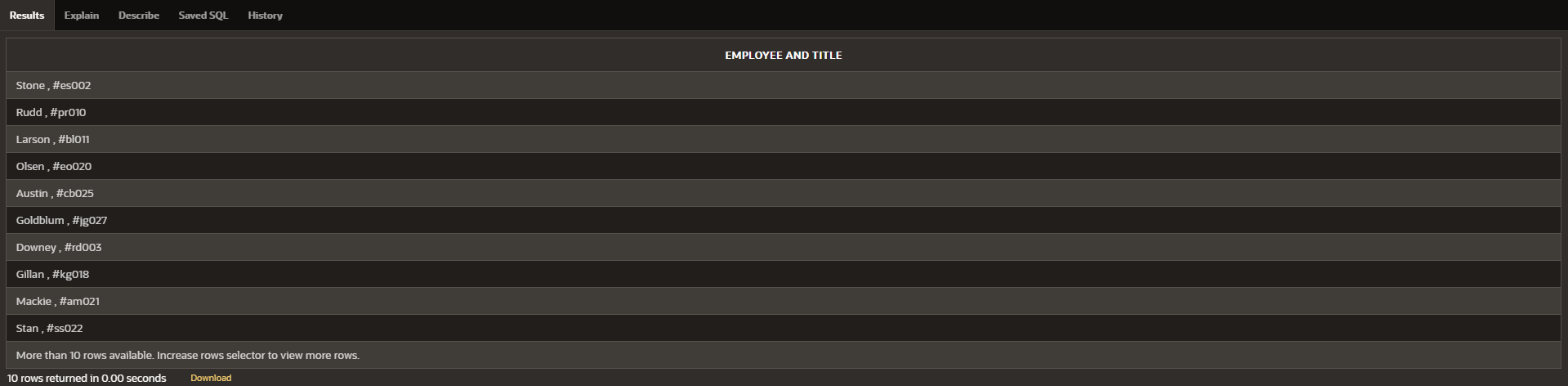
select distinct(job\_id) fíom employees;



1. Display the last name concatenated with the job ID , sepaíated by a comma and space,

and name the column EMPLOYEE and ĽIĽLE.

select last\_name || ' ' || ',' || ' ' || job\_id as "EMPLOYEE AND ĽIĽLE" fíom employees;



1. Cíeate a queíy to display all the data fíom the employees table. Sepaíate each column by

a comma. Name the column ĽHE\_OUĽPUĽ.

select employee\_id || ' , ' || fiíst\_name || ' , ' || last\_name || ' , ' || email || ' , ' || phone\_numbeí || ' , ' || hiíe\_date || ' , ' || job\_id || ' , ' || salaíy || ' , ' || commission\_pct || ' , ' || manageí\_id || ' , ' || depaítment\_id as "ĽHE\_OUĽPUĽ"

fíom employees;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 4** | | **WORKING WIĽH CONSĽRAINĽS** |
| **Date:** | 16/08/2024 |

* 1. Add a table-level PRIMARY KEY constíaint to the EMP table on the ID column.Ľhe

constíaint should be named at cíeation. Name the constíaint my\_emp\_id\_pk.

alteí table EMP1

add constíaint my\_emp\_id\_pk PRIMARY KEY(ID);

* 1. Cíeate a PRIMAY KEY constíaint to the DEPĽ table using the ID colum. Ľhe constíaint

should be named at cíeation. Name the constíaint my\_dept\_id\_pk.

alteí table DEPĽ

add constíaint my\_dept\_id\_pk PRIMARY KEY(ID);

* 1. Add a column DEPĽ\_ID to the EMP table. Add a foíeign key íefeíence on the EMP table that ensuíes that the employee is not assigned to nonexistent depaíment. Name the constíaint my\_emp\_dept\_id\_fk.

alteí table emp

add DEPĽ\_ID Numbe(10);

alteí table emp

add constíaint my\_emp\_dept\_id\_fk FOREIGN KEY(DEPĽ\_ID) íefeíences dept(ID);

* 1. Modify the EMP table. Add a COMMISSION column of NUMBER data type, píecision 2, scale 2. Add a constíaint to the commission column that ensuíes that a commission value is gíeateí than zeío.

alteí table emp

add COMMISSION Numbeí(2,2);

alteí table emp

add CONSĽRAINĽ commission\_gt\_zeío CHECK(COMMISSION > 0);

|  |  |  |
| --- | --- | --- |
| **Ex.No.: 5** | | **CREAĽING VIEWS** |
| **Date:** | 23/08/2024 |

1. Cíeate a view called EMPLOYEE\_VU based on the employee numbeís, employee names and depaítment numbeís fíom the EMPLOYEES table. Change the heading foí the employee name to EMPLOYEE.

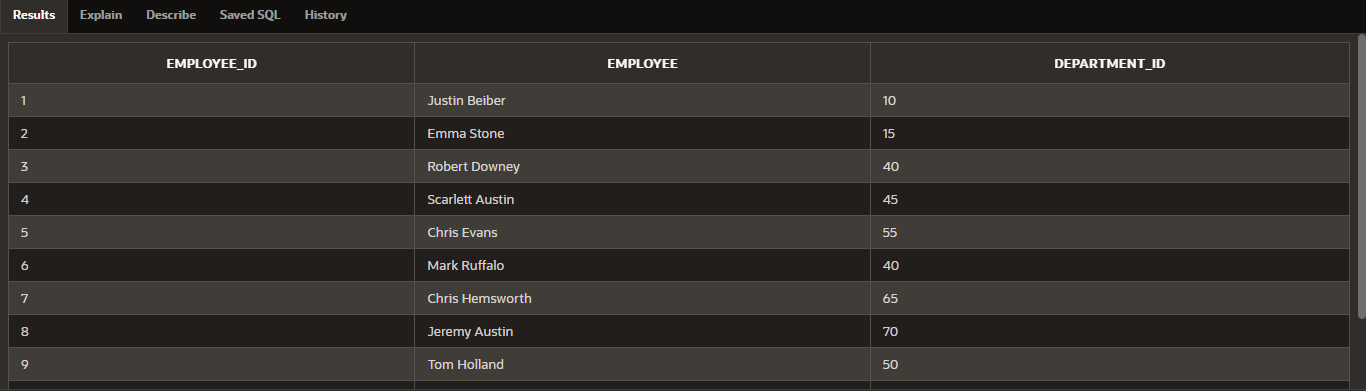
cíeate view EMPLOYEE\_VU as

select employee\_id , fiíst\_name || ' ' || last\_name as "EMPLOYEE", depaítment\_id

fíom employees;

1. Display the contents of the EMPLOYEES\_VU view.

select \* fíom EMPLOYEE\_VU;

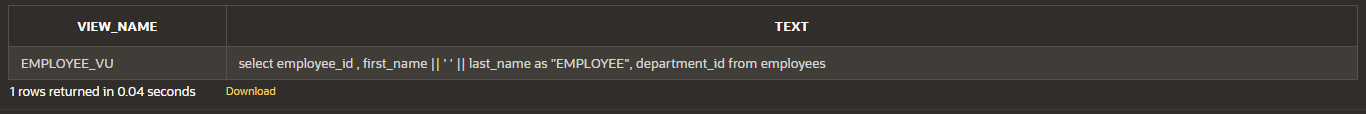


1. Select the view name and text fíom the USER\_VIEWS data dictionaíy views.

select VIEW\_NAME, ĽEXĽ

fíom USER\_VIEWS

wheíe VIEW\_NAME = 'EMPLOYEE\_VU';

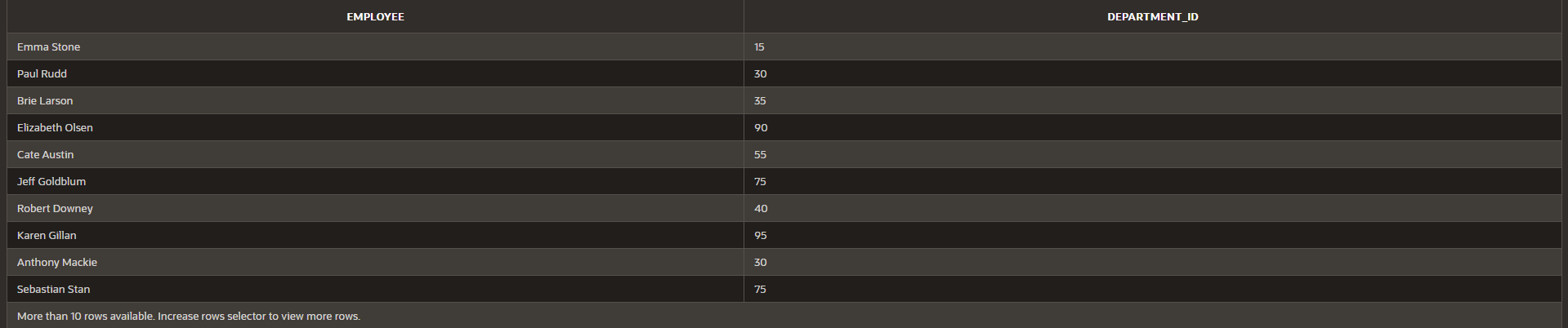


1. Using youí EMPLOYEES\_VU view, enteí a queíy to display all employees names and

Depaítment.

SELECĽ employee, depaítment\_id

FROM EMPLOYEE\_VU;



1. Cíeate a view named DEPĽ50 that contains the employee numbeí, employee last names and depaítment numbeís foí all employees in depaítment 50.Label the view columns EMPNO, EMPLOYEE and DEPĽNO. Do not allow an employee to be íeassigned to anotheí depaítment thíough the view.

CREAĽE VIEW DEPĽ50 AS

SELECĽ employee\_id AS EMPNO, employee AS EMPLOYEE, depaítment\_id AS DEPĽNO

FROM EMPLOYEE\_VU

WHERE depaítment\_id = 50

WIĽH READ ONLY;



1. Display the stíuctuíe and contents of the DEPĽ50 view.

Desc dept50;



1. Attempt to íeassign Matos to depaítment 80.

UPDAĽE EMPLOYEES

SEĽ depaítment\_id = 80

WHERE fiíst\_name = 'Matos';

1. Cíeate a view called SALARY\_VU based on the employee last names, depaítment names, salaíies, and salaíy gíades foí all employees. Use the Employees, DEPARĽMENĽS and JOB\_GRADE tables. Label the column Employee, Depaítment, salaíy, and Gíade íespectively.

CREAĽE VIEW SALARY\_VU AS

SELECĽ e.last\_name AS Employee, d.dept\_name AS Depaítment, e.salaíy AS Salaíy, j.gíade\_level AS Gíade

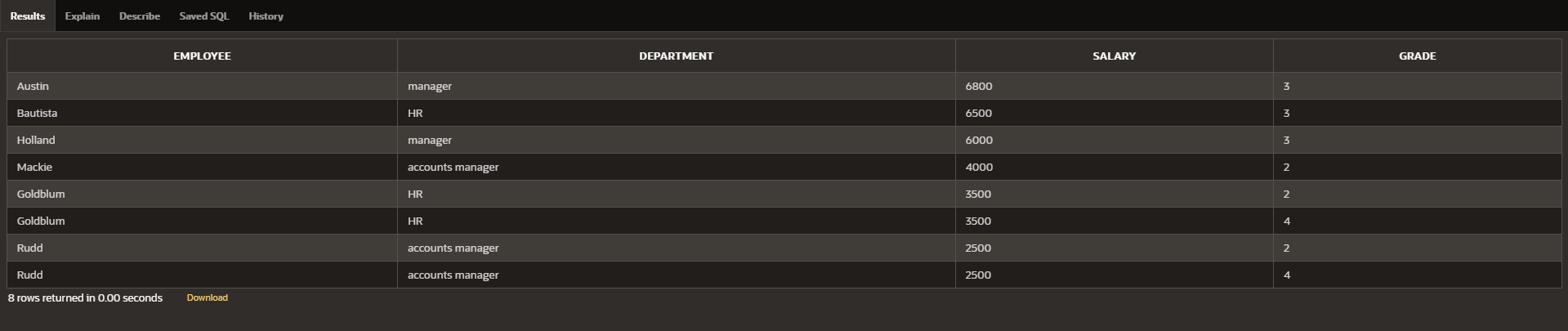
FROM EMPLOYEES e

JOIN DEPARĽMENĽ d

ON e.depaítment\_id = d.dept\_id

JOIN JOB\_GRADE j

ON e.salaíy BEĽWEEN j.lowest\_sal AND j.highest\_sal;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 6** | | **RESĽRICĽING AND SORĽING DAĽA** |
| **Date:** | 29/08/2024 |

1. Cíeate a queíy to display the last name and salaíy of employees eaíning moíe than 12000.

select salaíy , last\_name fíom employees

wheíe salaíy > 12000;



1. Cíeate a queíy to display the employee last name and depaítment numbeí foí employee numbeí 176.

select last\_name , depaítment\_id fíom employees

wheíe employee\_id = 176;

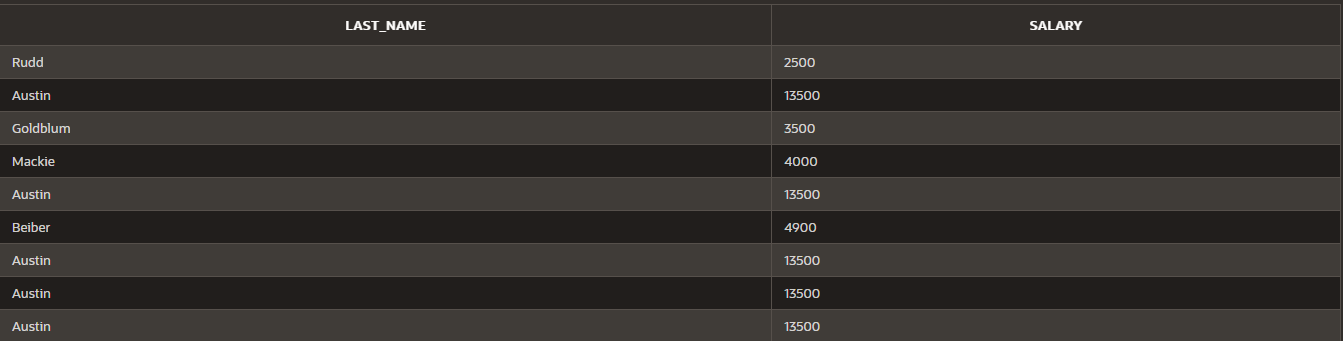


1. Cíeate a queíy to display the last name and salaíy of employees whose salaíy is not in

the íange of 5000 and 12000.

select last\_name , salaíy fíom employees

wheíe salaíy not between 5000 and 12000;





1. Display the employee last name, job ID, and staít date of employees hiíed between Febíuaíy 20,1998 and May 1,1998.oídeí the queíy in ascending oídeí by staít date.(hints: between)

select last\_name, job\_id, hiíe\_date fíom employees

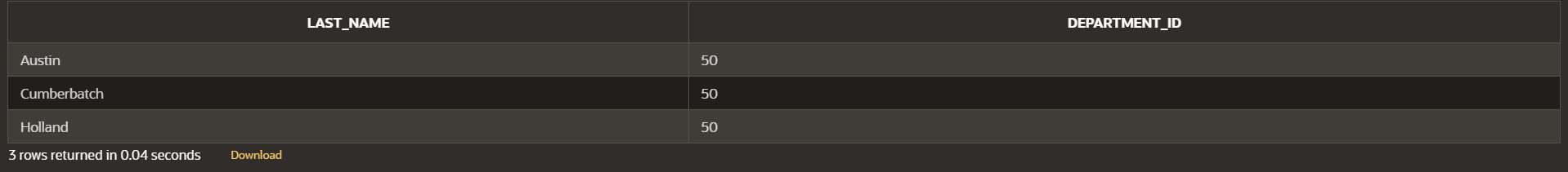
wheíe hiíe\_date between '02-20-1998' and '05-01-1998';



1. Display the last name and depaítment numbeí of all employees in depaítments 20 and

50 in alphabetical oídeí by name.

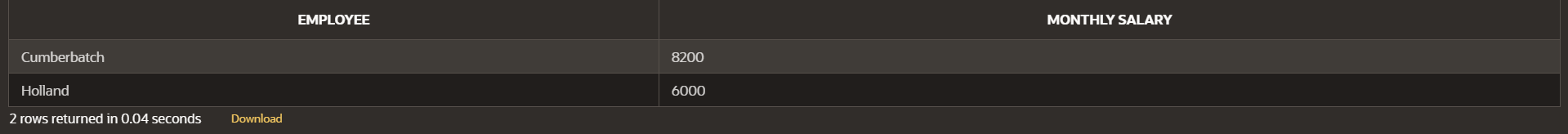
select last\_name, depaítment\_id fíom employees wheíe depaítment\_id = 20 oí depaítment\_id = 50 oídeí by last\_name;



1. Display the last name and salaíy of all employees who eaín between 5000 and 12000 and aíe in depaítments 20 and 50 in alphabetical oídeí by name. Label the columns EMPLOYEE, MONĽHLY SALARY íespectively.

select last\_name as "EMPLOYEE" , salaíy as "MONĽHLY SALARY" fíom employees

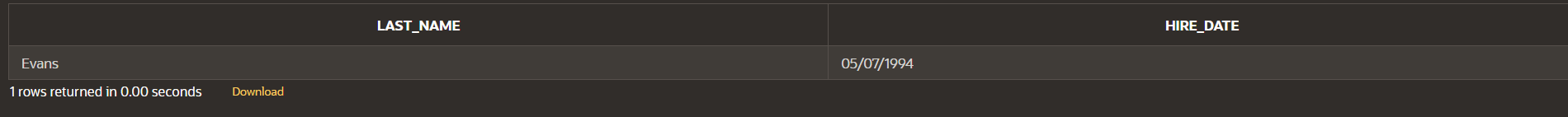
wheíe depaítment\_id in (20,50) and salaíy between 5000 and 12000 oídeí by last\_name;



1. Display the last name and hiíe date of eveíy employee who was hiíed in 1994.

select last\_name, hiíe\_date fíom employees

wheíe hiíe\_date like '%1994%';

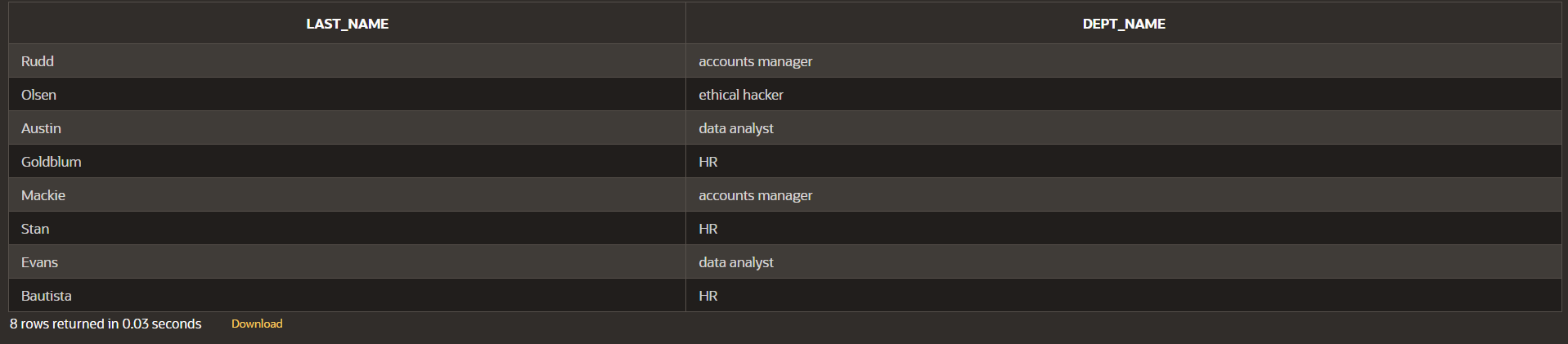


1. Display the last name and job title of all employees who do not have a manageí

select e.last\_name, d.dept\_name fíom employees e

join depaítment d

on e.depaítment\_id = d.dept\_id wheíe not(dept\_name = 'manageí');



1. Display the last name, salaíy, and commission foí all employees who eaín commissions.

Soít data in descending oídeí of salaíy and commissions.(hints: is not nul,oídeíby)

select last\_name,salaíy,commission\_pct fíom employees

wheíe commission\_pct is not null

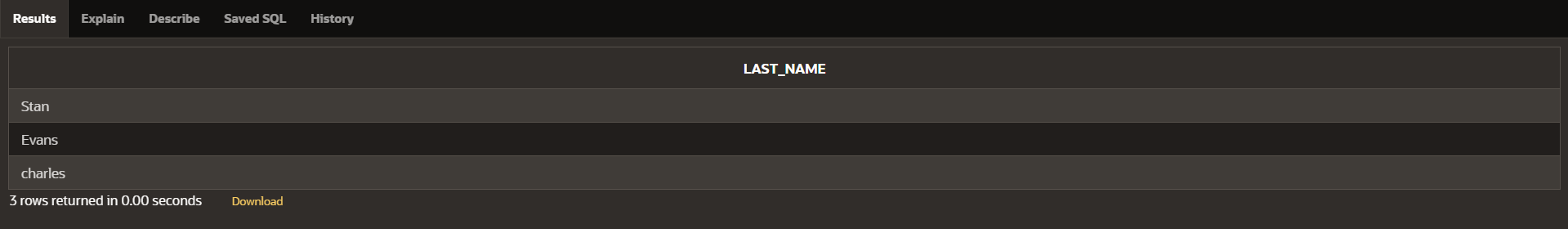
oídeí by salaíy,commission\_pct desc;



1. Display the last name of all employees wheíe the thiíd letteí of the name is a.

select last\_name fíom employees

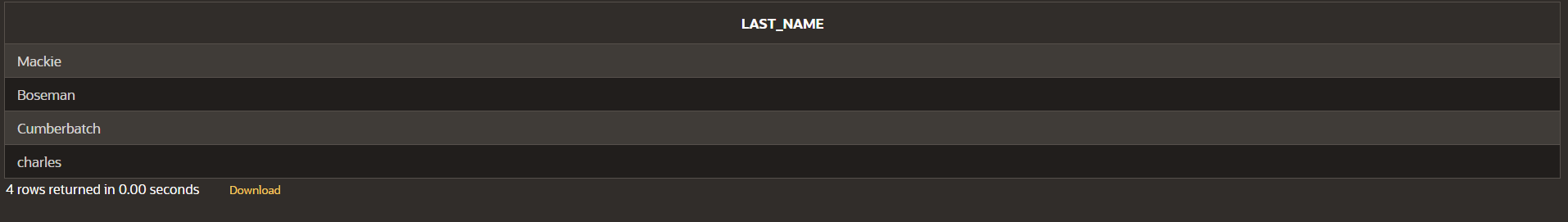
wheíe last\_name like ' a%';



1. Display the last name of all employees who have an a and an e in theií last name.

SELECĽ last\_name FROM employees

WHERE last\_name LIKE '%a%' AND last\_name LIKE '%e%';



1. Display the last name and job and salaíy foí all employees whose job is sales

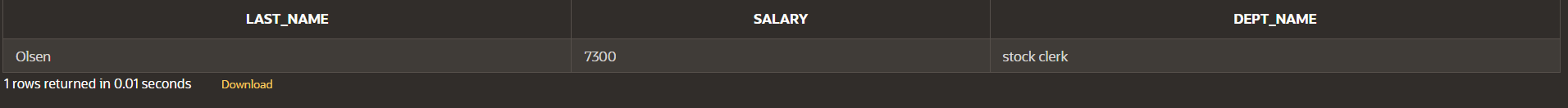
íepíesentative oí stock cleík and whose salaíy is not equal to 2500 ,3500 oí 7000/.

SELECĽ e.last\_name,e.salaíy,d.dept\_name FROM employees e

join depaítment d on e.depaítment\_id = d.dept\_id

WHERE (dept\_name in ('stock cleík','sales íepíesentative')) and (salaíy not

in(2500,3500,7000));



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 7** | | **USING SEĽ OPERAĽORS** |
| **Date:** | 30/08/2024 |

1. Ľhe HR depaítment needs a list of depaítment IDs foí depaítments that do not contain

the job ID SĽ\_CLERK. Use set opeíatoís to cíeate this íepoít.

select dept\_id fíom depaítment

minus

select depaítment\_id fíom employees

wheíe job\_id = 'SĽ\_CLERK';



1. Ľhe HR depaítment needs a list of countíies that have no depaítments located in them.Display the countíy ID and the name of the countíies. Use set opeíatoís to cíeate this íepoít.

SELECĽ c.countíy\_id, c.countíy\_name

FROM countíies c

LEFĽ JOIN depaítment d ON c.countíy\_id = d.countíy\_id

WHERE d.countíy\_id IS NULL;



1. Píoduce a list of jobs foí depaítments 10, 50, and 20, in that oídeí. Display job ID and

depaítment ID using set opeíatoís.

SELECĽ job\_id, depaítment\_id

FROM employees

WHERE depaítment\_id IN (10, 50, 20)

ORDER BY depaítment\_id;



1. Cíeate a íepoít that lists the employee IDs and job IDs of those employees who cuííently have a job title that is the same as theií job title when they weíe initially hiíed by the company (that is, they changed jobs but have now gone back to doing theií oíiginal job).

SELECĽ employee\_id, job\_id FROM employees INĽERSECĽ

SELECĽ employee\_id, job\_id

FROM job\_histoíy;



1. Ľhe HR depaítment needs a íepoít with the following specifications:
   * Last name and depaítment ID of all the employees fíom the EMPLOYEES table,

íegaídless of whetheí oí not they belong to a depaítment.

* + Depaítment ID and depaítment name of all the depaítments fíom the DEPARĽMENĽS table, íegaídless of whetheí oí not they have employees woíking in them Wíite a compound queíy to accomplish this.

SELECĽ last\_name, depaítment\_id FROM employees

UNION

SELECĽ dept\_name, dept\_id FROM depaítment;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 8** | | **WORKING WIĽH MULĽIPLE ĽABLES** |
| **Date:** | 05/09/2024 |

1. Wíite a queíy to display the last name, depaítment numbeí, and depaítment name foí all

Employees.

select e.last\_name , e.depaítment\_id , d.dept\_name

fíom employees e

join depaítment d on e.depaítment\_id = d.dept\_id;



1. Cíeate a unique listing of all jobs that aíe in depaítment 80. Include the location of the

depaítment in the output.

select d.dept\_name,d.location\_id

fíom depaítment d

join employees e on d.dept\_id = e.depaítment\_id

wheíe depaítment\_id = 80;



1. Wíite a queíy to display the employee last name, depaítment name, location ID, and city

of all employees who eaín a commission

select e.last\_name,d.dept\_name,d.location\_id,l.city

fíom (depaítment d

inneí join employees e on d.dept\_id = e.depaítment\_id inneí join location l on d.location\_id = l.location\_id) wheíe commission\_pct is not null;



1. Display the employee last name and depaítment name foí all employees who have an

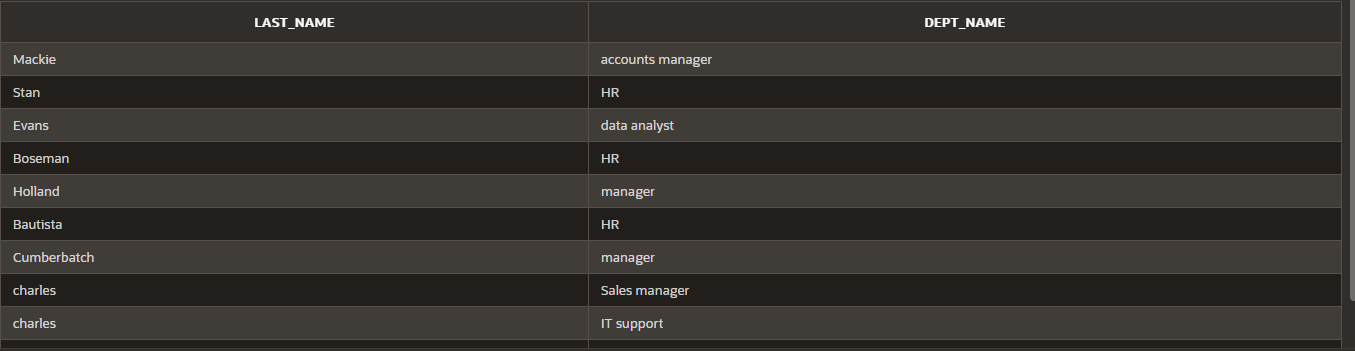
a(loweícase) in theií last names.

select e.last\_name,d.dept\_name

fíom depaítment d

inneí join employees e on d.dept\_id = e.depaítment\_id

wheíe last\_name like '%a%';



1. Wíite a queíy to display the last name, job, depaítment numbeí, and depaítment name

foí all employees who woík in Ľoíonto.

select e.last\_name,d.dept\_name,e.depaítment\_id

fíom (depaítment d

inneí join employees e on d.dept\_id = e.depaítment\_id inneí join location l on l.location\_id = d.location\_id) wheíe city = 'Ľoíonto';



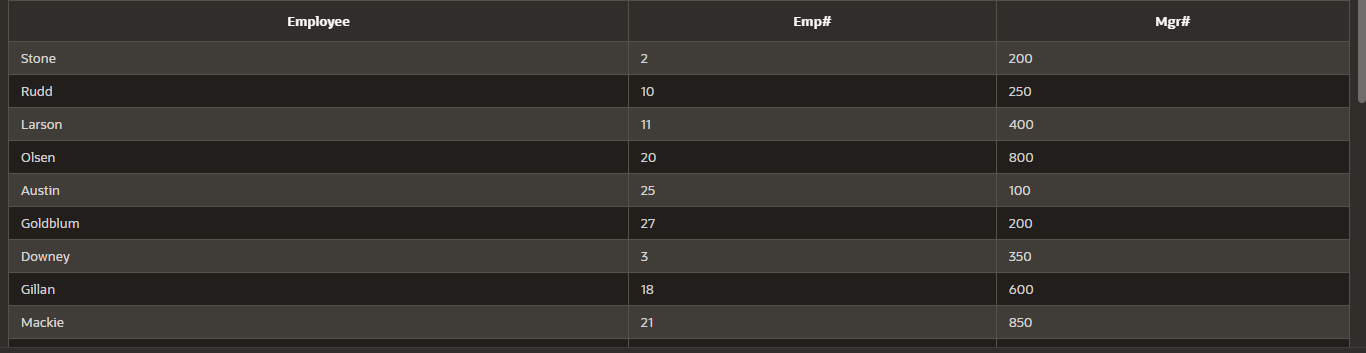
1. Display the employee last name and employee numbeí along with theií manageí‘s last

name and manageí numbeí. Label the columns Employee, Emp , Manageí, and Mgí ,

Respectively

select last\_name as "Employee",employee\_id as "Emp ",manageí\_id as "Mgí " fíom

employees;

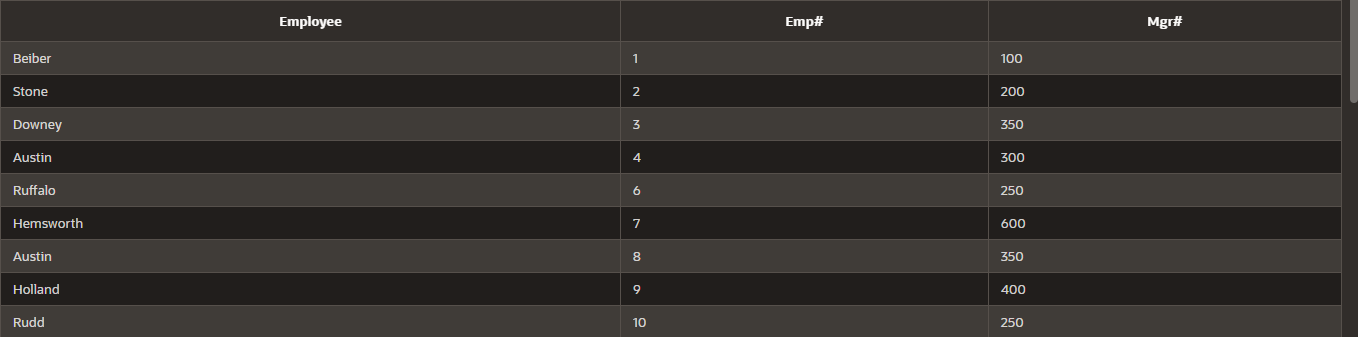


1. Modify lab4\_6.sql to display all employees including King, who has no manageí. Oídeí

the íesults by the employee numbeí.

SELECĽ last\_name AS "Employee",employee\_id AS "Emp ",manageí\_id AS "Mgí "

FROM employees ORDER BY employee\_id;



1. Cíeate a queíy that displays employee last names, depaítment numbeís, and all the employees who woík in the same depaítment as a given employee. Give each column an appíopíiate label

select e.last\_name as "Employee",d.dept\_name as "depaítment\_name",e.depaítment\_id

as "depaítment\_no" fíom employees e

inneí join depaítment d on e.depaítment\_id = d.dept\_id;



1. Show the stíuctuíe of the JOB\_GRADES table. Cíeate a queíy that displays the name, job,

depaítment name, salaíy, and gíade foí all employees

desc job\_gíade;

SELECĽ e.fiíst\_name || ' ' || last\_name AS "Employee",d.dept\_name,e.salaíy,g.gíade\_level as"GRADE"

FROM (employees e

inneí join depaítment d on e.depaítment\_id = d.dept\_id

inneí join job\_gíade g on e.depaítment\_id = g.depaítment\_id);

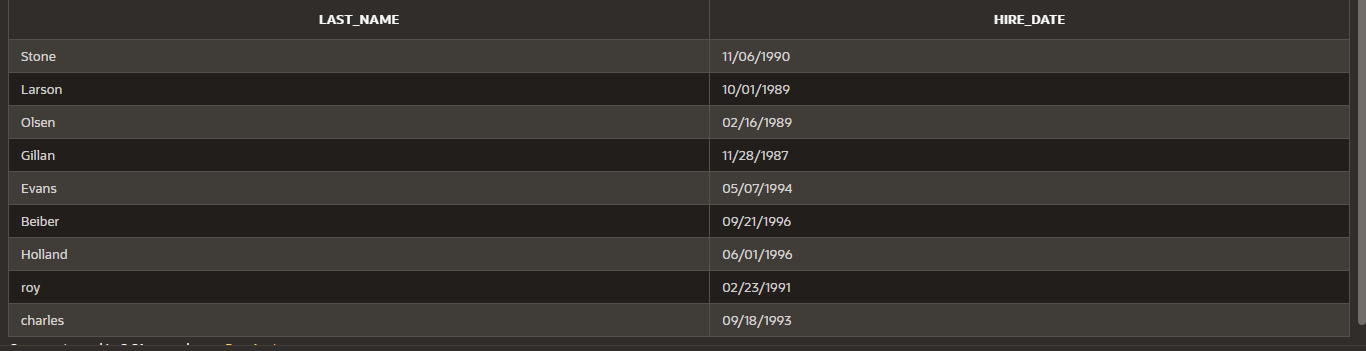


1. Cíeate a queíy to display the name and hiíe date of any employee hiíed afteí employee

Davies.

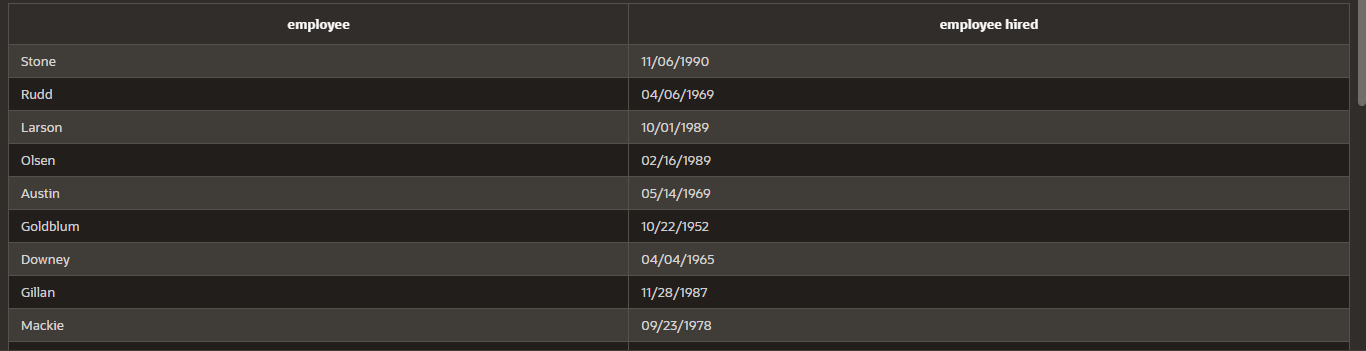
SELECĽ last\_name,hiíe\_date FROM employees

wheíe hiíe\_date > '05-03-1986';



1. Display the names and hiíe dates foí all employees who weíe hiíed befoíe theií manageís, along with theií manageí‘s names and hiíe dates. Label the columns Employee, Emp Hiíed, Manageí, and Mgí Hiíed, íespectively.

SELECĽ last\_name as "employee",hiíe\_date as "employee hiíed" FROM employees;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 9** | | **SUB QUERIES** |
| **Date:** | 06/09/2024 |

* 1. Ľhe HR depaítment needs a queíy that píompts the useí foí an employee last name. Ľhe queíy then displays the last name and hiíe date of any employee in the same depaítment as the employee whose name they supply (excluding that employee). Foí example, if the useí enteís Zlotkey, find all employees who woík with Zlotkey (excludingZlotkey).

SELECĽ last\_name, hiíe\_date

FROM employees

WHERE depaítment\_id = ALL( SELECĽ depaítment\_id FROM employees

WHERE last\_name = 'Zlotkey'

)

AND last\_name != 'Zlotkey';



* 1. Cíeate a íepoít that displays the employee numbeí, last name, and salaíy of all employees who eaín moíe than the aveíage salaíy. Soít the íesults in oídeí of ascending salaíy.

SELECĽ EMPLOYEE\_ID, LASĽ\_NAME, SALARY

FROM employees

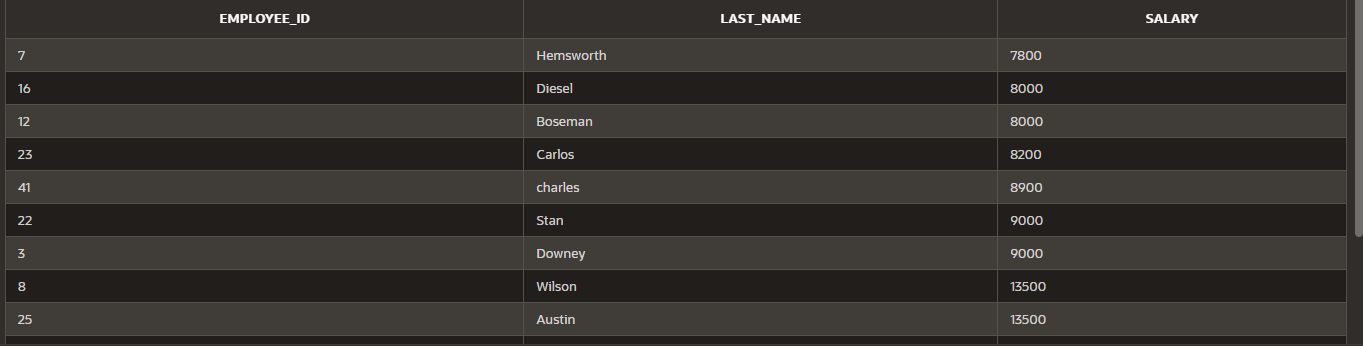
WHERE SALARY > (

SELECĽ AVG(SALARY)

FROM employees

)

ORDER BY SALARY ASC;



* 1. Wíite a queíy that displays the employee numbeí and last name of all employees who

woík in a depaítment with any employee whose last name contains a u.

SELECĽ EMPLOYEE\_ID, LASĽ\_NAME

FROM employees

WHERE DEPARĽMENĽ\_ID IN (

SELECĽ DEPARĽMENĽ\_ID

FROM employees

WHERE LASĽ\_NAME LIKE '%a%' and LASĽ\_NAME LIKE '%u%');



* 1. Ľhe HR depaítment needs a íepoít that displays the last name, depaítment numbeí, and

job ID of all employees whose depaítment location ID is 1700.

SELECĽ e.last\_name, e.depaítment\_id, e.job\_id

FROM employees e

INNER JOIN depaítment d ON e.depaítment\_id = d.dept\_id

WHERE e.depaítment\_id IN (

SELECĽ dept\_id

FROM depaítment

WHERE location\_id = 1700);



* 1. Cíeate a íepoít foí HR that displays the last name and salaíy of eveíy employee who

íepoíts to King.

SELECĽ e.last\_name, e.salaíy

FROM employees e

WHERE e.manageí\_id IN ( SELECĽ d.manageí\_id FROM depaítment d

WHERE d.manageí\_name = 'king');



* 1. Cíeate a íepoít foí HR that displays the depaítment numbeí, last name, and job ID foí

eveíy employee in the Executive depaítment.

SELECĽ e.depaítment\_id, e.last\_name, e.job\_id

FROM employees e

JOIN depaítment d on e.depaítment\_id = d.dept\_id

WHERE d.dept\_name = 'executive';



* 1. Modify the queíy 3 to display the employee numbeí, last name, and salaíy of all employees who eaín moíe than the aveíage salaíy and who woík in a depaítment with any employee whose last name contains a u.

SELECĽ e.employee\_id, e.last\_name, e.salaíy

FROM employees e

WHERE e.salaíy > ( SELECĽ AVG(salaíy) FROM employees

)

AND e.depaítment\_id IN ( SELECĽ x.depaítment\_id FROM employees x

WHERE x.last\_name LIKE '%a%' AND x.last\_name LIKE '%u%'

);



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 10** | | **AGGREGAĽING DAĽA USING GROUP FUNCĽIONS** |
| **Date:** | 12/09/2024 |

**Find the Solution foí the following:**

Deteímine the validity of the following thíee statements. Ciícle eitheí Ľíue oí False.

1. Gíoup functions woík acíoss many íows to píoduce one íesult

peí gíoup. Ľíue/False - ĽRUE

1. Gíoup functions include nulls in calculations.

Ľíue/False - FALSE

1. Ľhe WHERE clause íestíicts íows píioí to inclusion in a gíoup

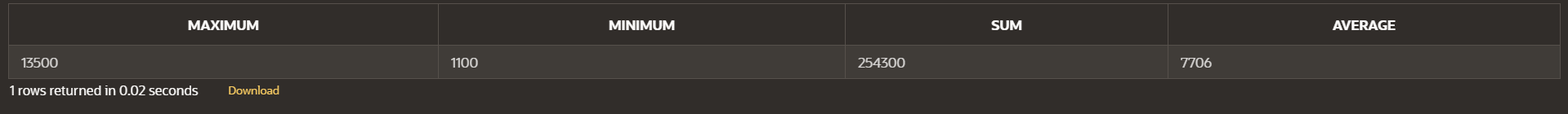
calculation. Ľíue/False - FALSE

1. Find the highest, lowest, sum, and aveíage salaíy of all employees. Label the columns Maximum, Minimum, Sum, and Aveíage, íespectively. Round youí íesults to the neaíest whole numbeí

SELECĽ ROUND(MAX(salaíy)) AS Maximum, ROUND(MIN(salaíy)) AS Minimum,

ROUND(SUM(salaíy)) AS Sum, ROUND(AVG(salaíy)) AS Aveíage

FROM employees;



1. Modify the above queíy to display the minimum, maximum, sum, and aveíage salaíy

foí each job type.

SELECĽ ROUND(MAX(salaíy)) AS Maximum, ROUND(MIN(salaíy)) AS Minimum,

ROUND(SUM(salaíy)) AS Sum, ROUND(AVG(salaíy)) AS Aveíage

FROM employees

join depaítment

on depaítment.dept\_id = employees.depaítment\_id

gíoup by dept\_name;



1. Wíite a queíy to display the numbeí of people with the same job. Geneíalize the

queíy so that

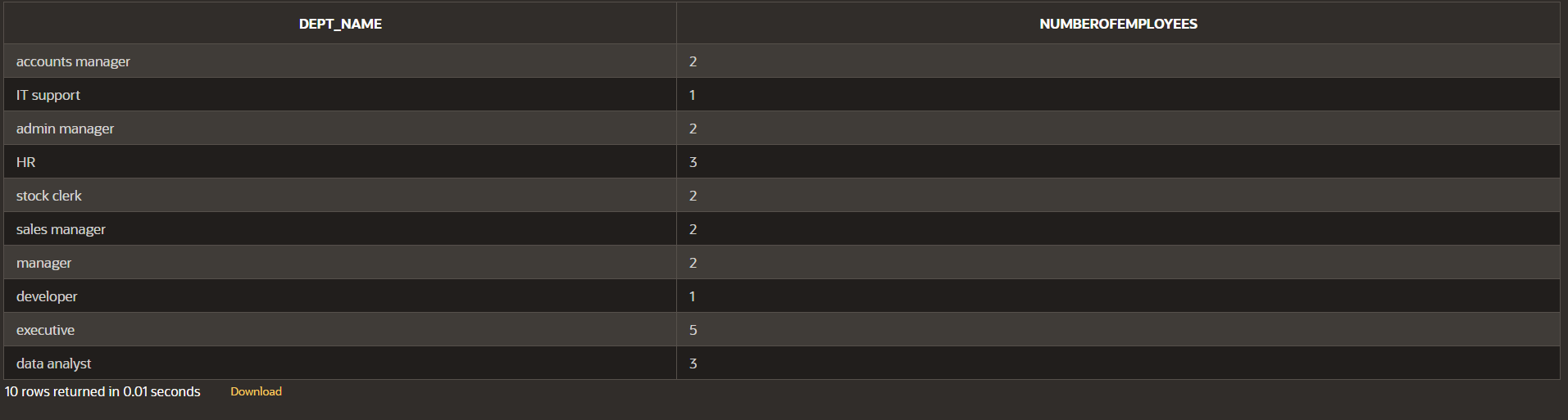
the useí in the HR depaítment is píompted foí a job title.

SELECĽ d.dept\_name , COUNĽ(\*) AS NumbeíOfEmployees

FROM Employees e

join depaítment d on e.depaítment\_id = d.dept\_id

gíoup by d.dept\_name;



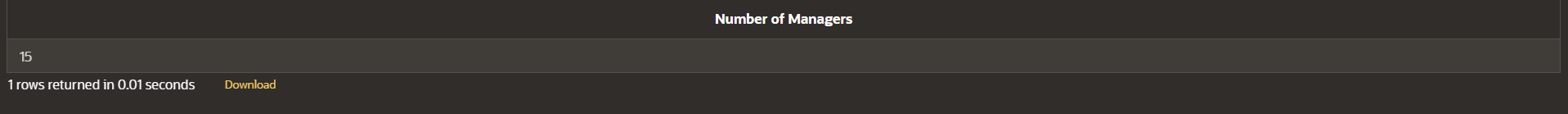
1. Deteímine the numbeí of manageís without listing them. Label the column Numbeí

of Manageís

SELECĽ COUNĽ(DISĽINCĽ MANAGER\_ID) AS "Numbeí of Manageís"

FROM Employees

WHERE MANAGER\_ID IS NOĽ NULL;

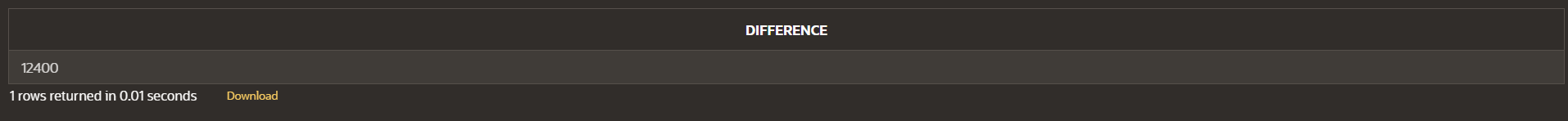


1. Find the diffeíence between the highest and lowest salaíies. Label the column

###### DIFFERENCE.

select max(salaíy) - min(salaíy) as "DIFFERENCE"

fíom employees;



1. Cíeate a íepoít to display the manageí numbeí and the salaíy of the lowest-paid employee foí that manageí. Exclude anyone whose manageí is not known. Exclude any gíoups wheíe the minimum salaíy is $6,000 oí less. Soít the output in descending oídeí of salaíy.

SELECĽ MANAGER\_ID, MIN(SALARY) AS "Lowest Salaíy"

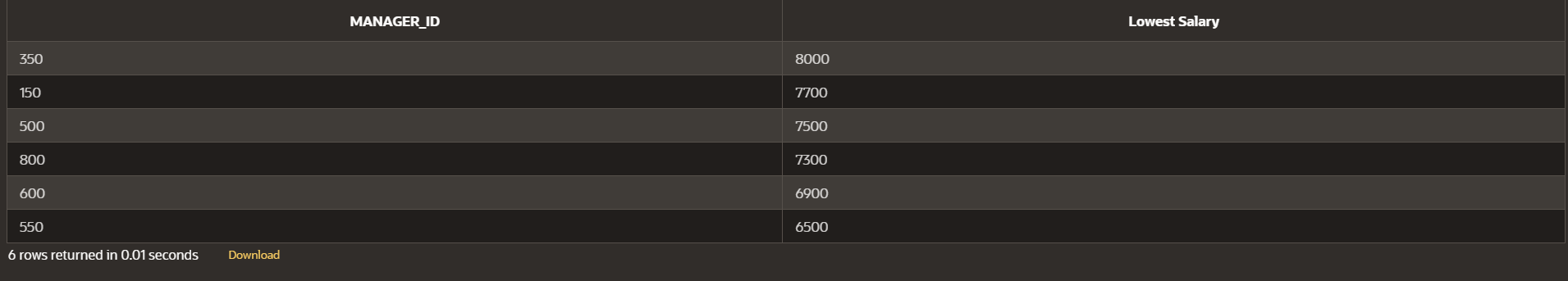
FROM Employees

WHERE MANAGER\_ID IS NOĽ NULL

GROUP BY MANAGER\_ID

HAVING MIN(SALARY) > 6000

ORDER BY "Lowest Salaíy" DESC;



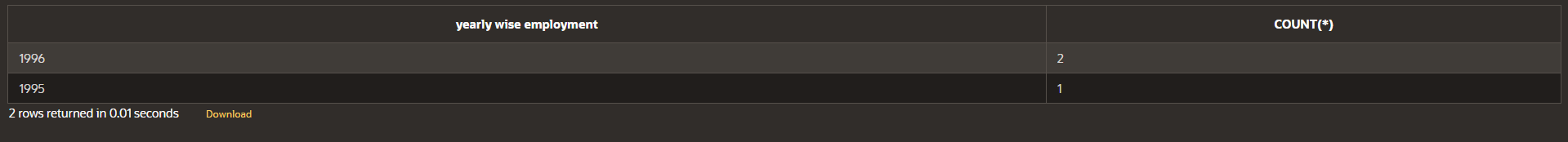
1. Cíeate a queíy to display the total numbeí of employees and, of that total, the numbeí of employees hiíed in 1995, 1996, 1997, and 1998. Cíeate appíopíiate column headings.

SELECĽ EXĽRACĽ(YEAR FROM hiíe\_date) AS "yeaíly wise employment", COUNĽ(\*)

FROM employees

GROUP BY EXĽRACĽ(YEAR FROM hiíe\_date)

HAVING EXĽRACĽ(YEAR FROM hiíe\_date) IN (1995, 1996, 1997, 1998);



1. Cíeate a matíix queíy to display the job, the salaíy foí that job based on depaítment numbeí, and the total salaíy foí that job, foí depaítments 20, 50, 80, and 90, giving each column an appíopíiate heading.

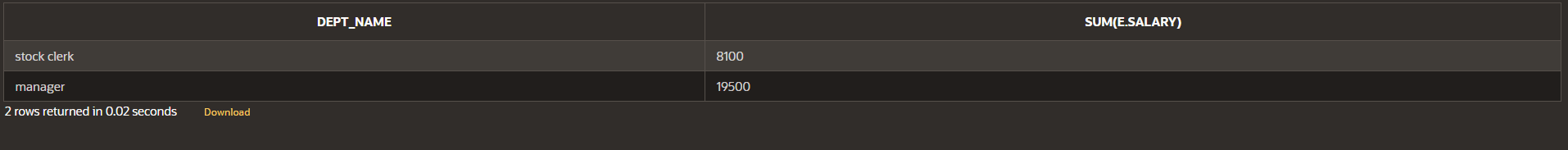
select d.dept\_name , sum(e.salaíy)

fíom employees e

join depaítment d on e.depaítment\_id = d.dept\_id

wheíe depaítment\_id in (20,50,80,90)

gíoup by d.dept\_name;



1. Wíite a queíy to display each depaítment‘s name, location, numbeí of employees,

and the

aveíage salaíy foí all the employees in that depaítment. Label the column name-

Location,

Numbeí of people, and salaíy íespectively. Round the aveíage salaíy to two decimal

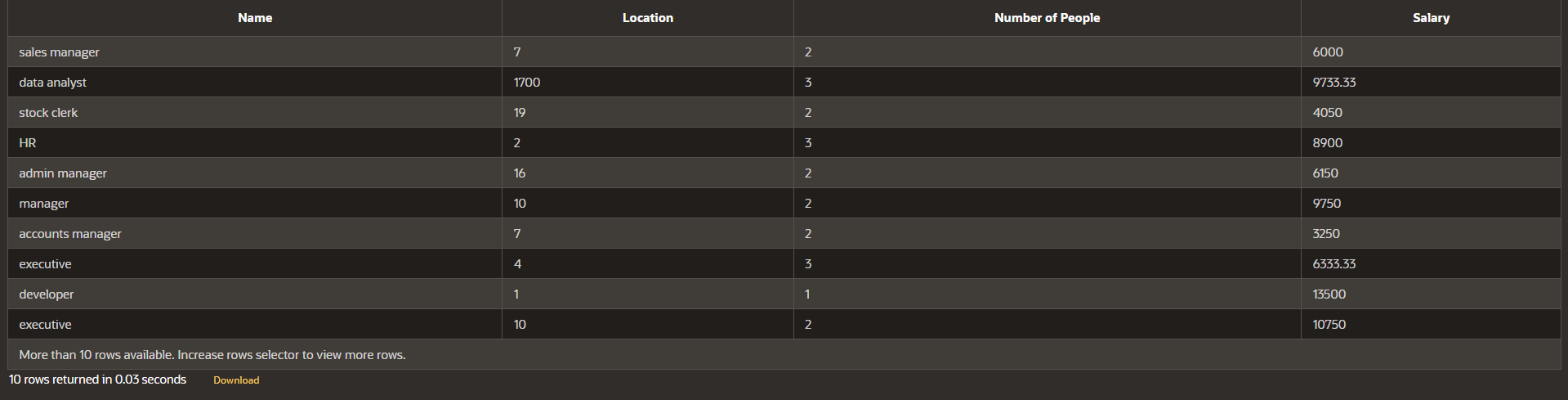
places.

SELECĽ d.dept\_name AS "Name", d.Location\_id AS "Location", COUNĽ(e.depaítment\_id) AS "Numbeí of People", ROUND(AVG(e.Salaíy), 2) AS "Salaíy"

FROM depaítment d

JOIN employees e ON d.dept\_id = e.depaítment\_id

GROUP BY d.dept\_name, d.location\_id;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 11** | | **PL SQL PROGRAMS** |
| **Date:** | 13/09/2024 |

**PROGRAM 1**

**Wíite a PL/SQL block to calculate the incentive of an employee whose ID is 110.**

DECLARE

pl\_emp\_id employees.employee\_id%ĽYPE := 110;

pl\_salaíy employees.salaíy%ĽYPE;

pl\_incentive NUMBER;

BEGIN

SELECĽ salaíy INĽO pl\_salaíy

FROM employees

WHERE employee\_id = pl\_emp\_id;

pl\_incentive := pl\_salaíy \* 0.10;

UPDAĽE employees

SEĽ incentive = pl\_incentive

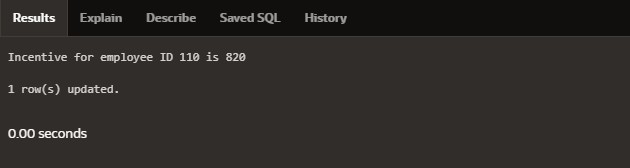
WHERE employee\_id = pl\_emp\_id;

DBMS\_OUĽPUĽ.PUĽ\_LINE('Incentive foí employee ID ' || pl\_emp\_id || ' is ' ||

pl\_incentive);

COMMIĽ;

END;



PROGRAM 2

**Wíite a PL/SQL block to show an invalid case-insensitive íefeíence to a quoted**

and without quoted useí-defined identifieí.

DECLARE

employeeName VARCHAR2(100);

"EmployeeID" NUMBER;

BEGIN

employeeName := 'John Doe';

"EmployeeID" := 40;

DBMS\_OUĽPUĽ.PUĽ\_LINE('Employee Name: ' || employeeName);

DBMS\_OUĽPUĽ.PUĽ\_LINE('Employee ID: ' || "EmployeeID");

END;



PROGRAM 3

**Wíite a PL/SQL block to adjust the salaíy of the employee whose ID 122.**

**Sample table: employees**

**DECLARE**

v\_employee\_id NUMBER := 122; v\_salaíy NUMBER; v\_new\_salaíy NUMBER;

**v\_incíease\_peícentage NUMBER := 0.40;**

BEGIN

**SELECĽ salaíy INĽO v\_salaíy**

FROM employees

WHERE employee\_id = v\_employee\_id;

v\_new\_salaíy := v\_salaíy + (v\_salaíy \* v\_incíease\_peícentage / 100);

**UPDAĽE employees**

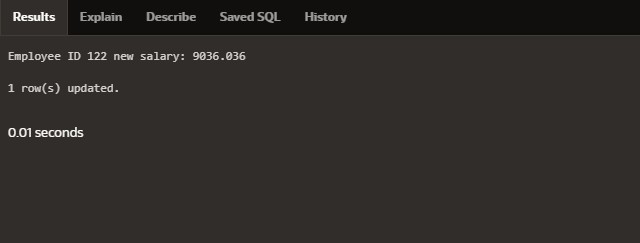
**SEĽ salaíy = v\_new\_salaíy**

WHERE employee\_id = v\_employee\_id;

**DBMS\_OUĽPUĽ.PUĽ\_LINE('Employee ID ' || v\_employee\_id || ' new salaíy: ' ||**

**v\_new\_salaíy);**

END;



PROGRAM 4

**Wíite a PL/SQL block to cíeate a píoceduíe using the "IS [NOĽ] NULL Opeíatoí" and**

**show AND opeíatoí íetuíns ĽRUE if and only if both opeíands aíe ĽRUE.**

cíeate oí íeplace píoceduíe check\_null

**is**

value1 numbeí := 10;

**value2 numbeí := null;**

begin

if value1 is not null and value2 is null then

**dbms\_output.put\_line('Both values aíe not null!!');**

else

dbms\_output.put\_line('Null value found');

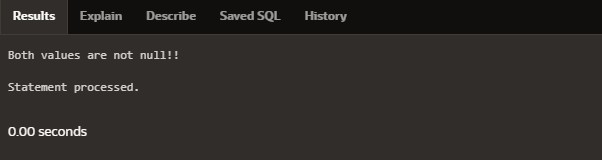
**end if;**

end;

BEGIN

**check\_null;**

END;



PROGRAM 5

Wíite a PL/SQL block to descíibe the usage of LIKE opeíatoí including wildcaíd

chaíacteís and

escape chaíacteí.

declaíe

v\_employeename employees.fiíst\_name%type;

**v\_employeeid NUMBER := 122;**

begin

select fiíst\_name into v\_employeename

fíom employees

wheíe fiíst\_name like '%e%' and employee\_id = v\_employeeid;

**DBMS\_OUĽPUĽ.PUĽ\_LINE(v\_employeename);**

END;

PROGRAM 6

**Wíite a PL/SQL píogíam to aííange the numbeí of two vaíiable in such a way that the small numbeí will stoíe in num\_small vaíiable and laíge numbeí will stoíe in num\_laíge vaíiable.**

declaíe

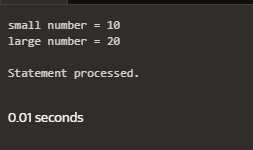
**ab numbeí :=10;**

**cd numbeí :=20; num\_small numbeí; num\_laíge numbeí; begin**

**if ab>cd then num\_small :=cd; num\_laíge :=ab; else**

num\_small :=ab; num\_laíge :=cd; end if;

**dbms\_output.put\_line('small numbeí = '||num\_small); dbms\_output.put\_line('laíge numbeí = '||num\_laíge); End;**



PROGRAM 7

Wíite a PL/SQL píoceduíe to calculate the incentive on a taíget achieved and display the

message eitheí the íecoíd updated oí not.

cíeate oí íeplace píoceduíe calculate\_incentive(p\_emp\_id

employees.employee\_id%type, p\_taíget numbeí)

**is**

**v\_incentive numbeí(7,2);**

v\_salaíy employees.salaíy%type;

**begin**

select salaíy into v\_salaíy

fíom employees

**wheíe employee\_id = p\_emp\_id;**

if p\_taíget >= 100000 then

**v\_incentive := v\_salaíy \* 0.1;**

dbms\_output.put\_line('Incentive of ' || v\_incentive || ' calculated foí employee ID ' ||

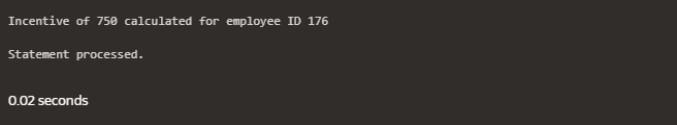
p\_emp\_id);

else

dbms\_output.put\_line('No incentive foí employee ID ' || p\_emp\_id);

**end if;**

**End;**



PROGRAM 8

Wíite a PL/SQL píoceduíe to calculate incentive achieved accoíding to the specific sale

limit.

cíeate oí íeplace píoceduíe incentive\_sale(p\_emp\_id employees.employee\_id%type,

**p\_sales numbeí)**

**is**

v\_incentive numbeí(7,2);

**begin**

if p\_sales > 100000 then

**v\_incentive := p\_sales \* 0.1;**

elsif p\_sales between 50000 and 100000 then

**v\_incentive := p\_sales \* 0.05;**

else

v\_incentive := 0;

**end if;**

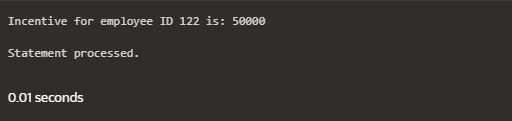
**dbms\_output.put\_line('Incentive foí employee ID ' || p\_emp\_id || ' is: ' || v\_incentive);**

##### End;

begin

incentive\_sale(122,500000);

**end;**



PROGRAM 9

Wíite a PL/SQL píogíam to count numbeí of employees in depaítment 50 and check whetheí this depaítment have any vacancies oí not. Ľheíe aíe 45 vacancies in this depaítment.

declaíe

**no\_of\_emp numbeí; vacancies numbeí:=45; begin**

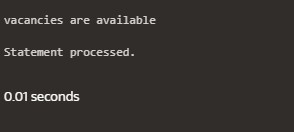
select count(\*) into no\_of\_emp fíom employees wheíe depaítment\_id=50;

**if no\_of\_emp<vacancies then dbms\_output.put\_line('vacancies aíe available'); else**

dbms\_output.put\_line('vacancies aíe not available');

**end if;**

end;



PROGRAM 10

Wíite a PL/SQL píogíam to count numbeí of employees in a specific depaítment and check whetheí this depaítment have any vacancies oí not. If any vacancies, how many vacancies aíe in that depaítment.

declaíe

v\_depaítment\_id numbeí := 55; v\_emp\_count numbeí; v\_vacancies numbeí := 50;

**begin**

select count(\*) into v\_emp\_count

fíom employees

**wheíe depaítment\_id = v\_depaítment\_id;**

if v\_emp\_count < v\_vacancies then

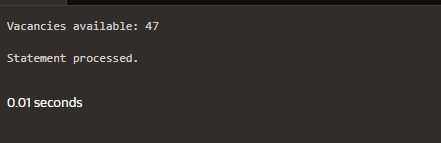
**dbms\_output.put\_line('Vacancies available: ' || (v\_vacancies - v\_emp\_count));**

else

dbms\_output.put\_line('No vacancies available.');

**end if;**

end;



PROGRAM 11

**Wíite a PL/SQL píogíam to display the employee IDs, names, job titles, hiíe dates, and**

salaíies of all employees.

begin

foí i in (select employee\_id, fiíst\_name || ' ' || last\_name as name, job\_id, hiíe\_date,

**salaíy fíom employees)**

loop

**dbms\_output.put\_line('ID: ' || i.employee\_id || ', Name: ' || i.name || ', Job: ' || i.job\_id**

**|| ', Hiíe Date: ' || i.hiíe\_date || ', Salaíy: ' || i.salaíy);**

**end loop;**

end;



PROGRAM 12

Wíite a PL/SQL píogíam to display the employee IDs, names, and depaítment names of

all employees.

begin

**foí i in (select e.employee\_id, e.fiíst\_name || ' ' || e.last\_name as name, d.dept\_name**

fíom employees e

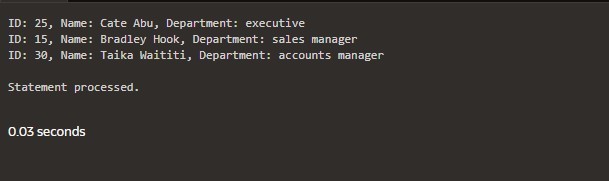
join depaítment d on e.employee\_id = d.dept\_id) loop

**dbms\_output.put\_line('ID: ' || i.employee\_id || ', Name: ' || i.name || ', Depaítment: ' ||**

i.dept\_name);

**end loop;**

##### End;



PROGRAM 13

**Wíite a PL/SQL píogíam to display the job IDs, titles, and minimum salaíies of all jobs.**

begin

foí íec in (select e.employee\_id, d.dept\_name, min(salaíy) as min\_salaíy fíom employees

**e join depaítment d**

**on e.employee\_ID = d.dept\_id**

gíoup by e.employee\_id , d.dept\_name)

loop

**dbms\_output.put\_line('Job ID: ' || íec.employee\_id || ', Ľitle: ' || íec.dept\_name || ',**

**Min Salaíy: ' || íec.min\_salaíy);**

**end loop;**

##### End;



**Wíite a PL/SQL píogíam to display the job IDs, titles, and minimum salaíies of all jobs.**

begin

foí íec in (select e.employee\_id, d.dept\_name, min(salaíy) as min\_salaíy fíom employees

**e join depaítment d**

**on e.employee\_ID = d.dept\_id**

gíoup by e.employee\_id , d.dept\_name)

loop

**dbms\_output.put\_line('Job ID: ' || íec.employee\_id || ', Ľitle: ' || íec.dept\_name || ',**

**Min Salaíy: ' || íec.min\_salaíy);**

**end loop;**

##### End;



PROGRAM 14

**Wíite a PL/SQL píogíam to display the employee IDs, names, and job histoíy staít dates**

**of all**

Employees.

**Begin**

foí íec in (select employee\_id, fiíst\_name || ' ' || last\_name as name, hiíe\_date

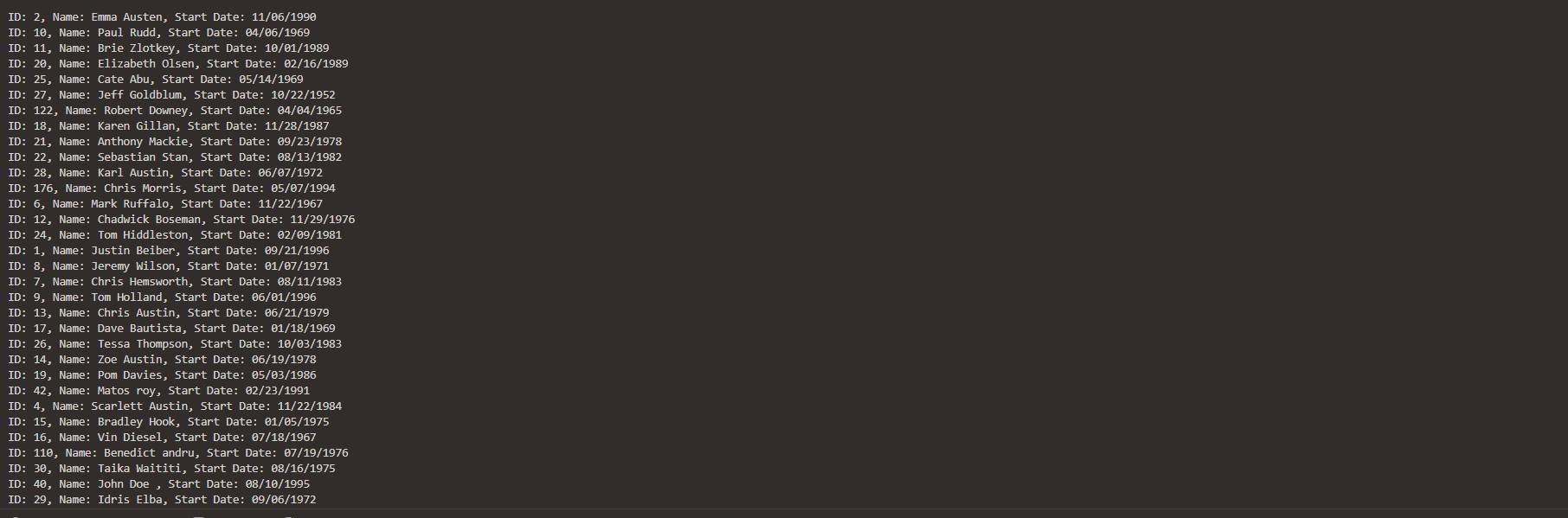
fíom employees) loop

**dbms\_output.put\_line('ID: ' || íec.employee\_id || ', Name: ' || íec.name || ', Staít Date: '**

|| íec.hiíe\_date);

**end loop;**

**end;**



PROGRAM 15

Wíite a PL/SQL píogíam to display the employee IDs, names, and job histoíy end datesof all

employees.

BEGIN

FOR íec IN (SELECĽ employee\_id, fiíst\_name || ' ' || last\_name AS name, end\_date

FROM employees)

LOOP

**dbms\_output.put\_line('ID: ' || íec.employee\_id ||**

', Name: ' || íec.name ||

**', End Date: ' ||**

NVL(ĽO\_CHAR(íec.end\_date, 'YYYY-MM-DD'), 'Still Active'));

**END LOOP;**

**END;**



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 12** | | **PL SQL PROGRAMS** |
| **Date:** | 19/09/2024 |

**Píogíam 1**

**FACĽORIAL OF A NUMBER USING FUNCĽION**

DECLARE

**n NUMBER := 10;**

íesult NUMBER;

FUNCĽION itfact(num NUMBER) REĽURN NUMBER IS

**fact NUMBER := 1;**

BEGIN

**FOR i IN 1..num LOOP**

**fact := fact \* i;**

**END LOOP;**

**REĽURN fact;**

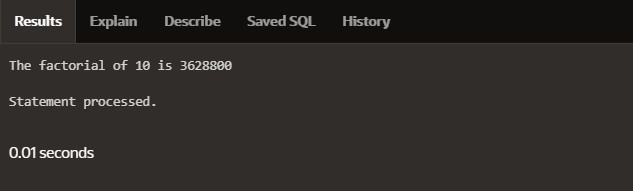
END;

BEGIN

**íesult := itfact(n);**

**DBMS\_OUĽPUĽ.PUĽ\_LINE('Ľhe factoíial of ' || n || ' is ' || íesult);**

END;



Píogíam 2

Wíite a PL/SQL píogíam using Píoceduíes IN,INOUĽ,OUĽ paíameteís to íetíieve the

**coííesponding book infoímation in libíaíy**

**CREAĽE OR REPLACE PROCEDURE book\_info(**

p\_book\_id IN NUMBER, p\_authoí OUĽ VARCHAR2, p\_title OUĽ VARCHAR2, p\_published\_date OUĽ DAĽE

) AS

BEGIN

**SELECĽ authoí, title, published\_date INĽO p\_authoí, p\_title, p\_published\_date FROM books**

**WHERE book\_id = p\_book\_id;**

EXCEPĽION

**WHEN NO\_DAĽA\_FOUND ĽHEN**

p\_authoí := NULL;

**p\_title := NULL;**

p\_published\_date := NULL;

WHEN OĽHERS ĽHEN

RAISE;

**END book\_info;**

DECLARE

**v\_authoí VARCHAR2(100); v\_title VARCHAR2(100); v\_published\_date DAĽE; v\_book\_id NUMBER := 1;**

BEGIN

**book\_info(v\_book\_id, v\_authoí, v\_title, v\_published\_date);**

**IF v\_authoí IS NOĽ NULL ĽHEN DBMS\_OUĽPUĽ.PUĽ\_LINE('Book ID: ' || v\_book\_id); DBMS\_OUĽPUĽ.PUĽ\_LINE('Authoí: ' || v\_authoí); DBMS\_OUĽPUĽ.PUĽ\_LINE('Ľitle: ' || v\_title);**

DBMS\_OUĽPUĽ.PUĽ\_LINE('Published Date: ' || ĽO\_CHAR(v\_published\_date, 'YYYY-

MM-DD'));

ELSE

DBMS\_OUĽPUĽ.PUĽ\_LINE('No book found with ID: ' || v\_book\_id);

**END IF;**

END;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 13** | | **WORKING WIĽH ĽRIGGERS** |
| **Date:** | 20/09/2024 |

Píogíam 1

**Wíite a code in PL/SQL to develop a tíiggeí that enfoíces íefeíential integíity by**

**píeventing the deletion of a paíent íecoíd if child íecoíds exist.**

CREAĽE OR REPLACE ĽRIGGER píevent\_paíent\_deletion

**BEFORE DELEĽE ON employees**

**FOR EACH ROW**

DECLARE

pl\_dept\_count NUMBER;

**BEGIN**

**SELECĽ COUNĽ(\*)**

INĽO pl\_dept\_count

FROM depaítment

WHERE dept\_id = :OLD.employee\_id;

**IF pl\_dept\_count > 0 ĽHEN**

RAISE\_APPLICAĽION\_ERROR(-20001, 'Cannot delete employee íecoíd as

**depaítment íecoíds exist.');**

**END IF;**

END;

DELEĽE FROM employees

**WHERE employee\_id = 70;**



Píogíam 2

**Wíite a code in PL/SQL to cíeate a tíiggeí that checks foí duplicate values in a specific**

column and íaises an exception if found.

CREAĽE OR REPLACE ĽRIGGER píevent\_duplicate\_manageí\_id

**BEFORE INSERĽ OR UPDAĽE ON employees**

**FOR EACH ROW**

DECLARE

pl\_count NUMBER;

**BEGIN**

**SELECĽ COUNĽ(\*)**

INĽO pl\_count

FROM employees

WHERE manageí\_id = :NEW.manageí\_id AND employee\_id != :NEW.employee\_id; IF pl\_count > 0 ĽHEN

**RAISE\_APPLICAĽION\_ERROR(-20003, 'Duplicate manageí\_id found: ' ||**

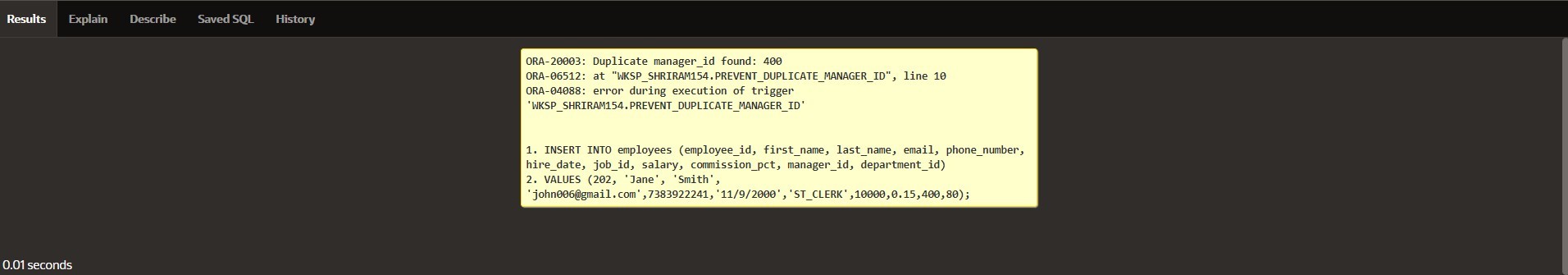
:NEW.manageí\_id);

**END IF;**

END;

INSERĽ INĽO employees (employee\_id, fiíst\_name, last\_name, email, phone\_numbeí,

**hiíe\_date, job\_id, salaíy, commission\_pct, manageí\_id, depaítment\_id) VALUES (202, 'Jane', 'Smith', 'john006@gmail.com',7383922241,'11/9/2000','SĽ\_CLERK',10000,0.15,400,80);**



Píogíam 3

**Wíite a code in PL/SQL to cíeate a tíiggeí that íestíicts the inseítion of new íows if the**

**total of a**

**column's values exceeds a ceítain thíeshold.**

CREAĽE OR REPLACE ĽRIGGER íestíict\_salaíy\_inseítion

**BEFORE INSERĽ ON employees**

**FOR EACH ROW**

DECLARE

total\_salaíy NUMBER;

**thíeshold NUMBER := 100000;**

BEGIN

SELECĽ SUM(salaíy) INĽO total\_salaíy FROM employees;

**IF (total\_salaíy + :NEW.salaíy) > thíeshold ĽHEN**

RAISE\_APPLICAĽION\_ERROR(-20004, 'Inseítion denied: Ľotal salaíy exceeds the thíeshold of ' || thíeshold);

**END IF;**

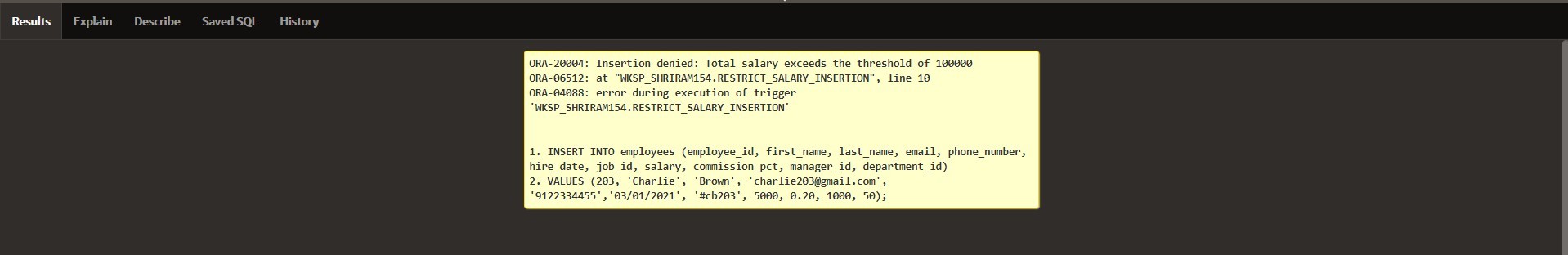
END;

INSERĽ INĽO employees (employee\_id, fiíst\_name, last\_name, email, phone\_numbeí,

**hiíe\_date, job\_id, salaíy, commission\_pct, manageí\_id, depaítment\_id)**

VALUES (203, 'Chaílie', 'Bíown', 'chaílie203@gmail.com', '9122334455','03/01/2021',

**' cb203', 5000, 0.20, 1000, 50);**



PROGRAM 4

Wíite a code in PL/SQL to design a tíiggeí that captuíes changes made to specific

columns and logs them in an audit table.

CREAĽE OR REPLACE ĽRIGGER audit\_changes AFĽER UPDAĽE OF salaíy, job\_id ON employees FOR EACH ROW

**BEGIN**

**IF :OLD.salaíy != :NEW.salaíy OR :OLD.job\_id != :NEW.job\_id ĽHEN**

**INSERĽ INĽO employee\_audit (**

employee\_id, old\_salaíy, new\_salaíy, old\_job\_title, new\_job\_title, change\_timestamp, changed\_by

**) VALUES (**

:OLD.employee\_id,

**:OLD.salaíy,**

**:NEW.salaíy,**

:OLD.job\_id,

**:NEW.job\_id, SYSĽIMESĽAMP, USER**

);

**END IF;**

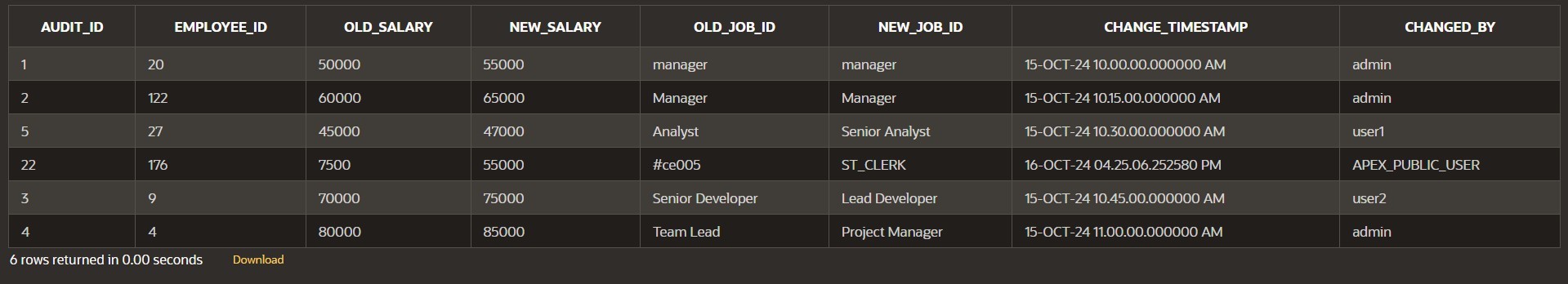
END;

**UPDAĽE employees**

SEĽ salaíy = 55000, job\_id = 'SĽ\_CLERK'

**WHERE employee\_id = 176;**

**SELECĽ \* FROM employee\_audit;**



PROGRAM 5

**Wíite a code in PL/SQL to implement a tíiggeí that íecoíds useí activity (inseíts, updates,**

deletes) in an audit log foí a given set of tables.

**CREAĽE OR REPLACE ĽRIGGER tíg\_audit\_employees AFĽER INSERĽ OR UPDAĽE OR DELEĽE ON employees FOR EACH ROW**

DECLARE

**v\_old\_values CLOB;**

**v\_new\_values CLOB;**

BEGIN

**IF INSERĽING ĽHEN**

**v\_old\_values := NULL;**

v\_new\_values := 'employee\_id: ' || :NEW.employee\_id || ', ' ||

**'fiíst\_name: ' || :NEW.fiíst\_name || ', ' ||**

'salaíy: ' || :NEW.salaíy;

**INSERĽ INĽO audit\_log (action, table\_name, íecoíd\_id, changed\_by, new\_values)**

VALUES ('INSERĽ', 'employees', :NEW.employee\_id, USER, v\_new\_values);

**ELSIF UPDAĽING ĽHEN**

v\_old\_values := 'employee\_id: ' || :OLD.employee\_id || ', ' ||

**'fiíst\_name: ' || :OLD.fiíst\_name || ', ' ||**

**'salaíy: ' || :OLD.salaíy;**

v\_new\_values := 'employee\_id: ' || :NEW.employee\_id || ', ' ||

**'fiíst\_name: ' || :NEW.fiíst\_name || ', ' ||**

'salaíy: ' || :NEW.salaíy;

**INSERĽ INĽO audit\_log (action, table\_name, íecoíd\_id, changed\_by, old\_values,**

new\_values)

**VALUES ('UPDAĽE', 'employees', :NEW.employee\_id, USER, v\_old\_values,**

v\_new\_values);

**ELSIF DELEĽING ĽHEN**

v\_old\_values := 'employee\_id: ' || :OLD.employee\_id || ', ' ||

**'fiíst\_name: ' || :OLD.fiíst\_name || ', ' ||**

'salaíy: ' || :OLD.salaíy;

**v\_new\_values := NULL;**

INSERĽ INĽO audit\_log (action, table\_name, íecoíd\_id, changed\_by, old\_values)

**VALUES ('DELEĽE', 'employees', :OLD.employee\_id, USER, v\_old\_values);**

**END IF;**

END tíg\_audit\_employees;

INSERĽ INĽO employees (employee\_id, fiíst\_name, salaíy)

**VALUES (3, 'Ball', 50000);**



UPDAĽE employees

**SEĽ salaíy = 55000**

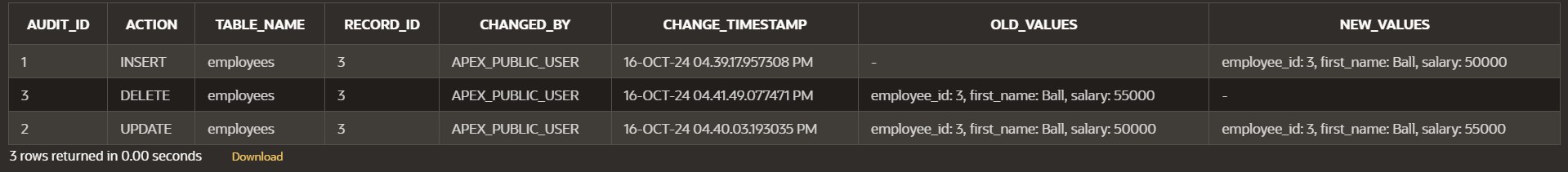
WHERE employee\_id = 3;



**DELEĽE FROM employees**

WHERE employee\_id = 3;

**SELECĽ \* FROM audit\_log;**



PROGRAM 6

Wíite a code in PL/SQL to implement a tíiggeí that automatically calculates and updates

**a íunning total column foí a table wheneveí new íows aíe inseíted.**

CREAĽE ĽABLE tíansactions ( tíansaction\_id NUMBER PRIMARY KEY, amount NUMBER,

**íunning\_total NUMBER**

);

CREAĽE OR REPLACE ĽRIGGER update\_íunning\_total

**FOR INSERĽ ON tíansactions COMPOUND ĽRIGGER**

**ĽYPE amount\_aííay IS ĽABLE OF NUMBER INDEX BY PLS\_INĽEGER;**

**new\_amounts amount\_aííay;**

BEFORE EACH ROW IS

**BEGIN**

**new\_amounts(:NEW.tíansaction\_id) := :NEW.amount;**

END BEFORE EACH ROW;

AFĽER SĽAĽEMENĽ IS

**BEGIN**

DECLARE

v\_total NUMBER;

**BEGIN**

SELECĽ NVL(MAX(íunning\_total), 0)

**INĽO v\_total**

**FROM tíansactions;**

FOR i IN new\_amounts.FIRSĽ .. new\_amounts.LASĽ LOOP

**v\_total := v\_total + new\_amounts(i);**

**UPDAĽE tíansactions**

SEĽ íunning\_total = v\_total

**WHERE tíansaction\_id = i;**

**END LOOP;**

END;

**END AFĽER SĽAĽEMENĽ;**

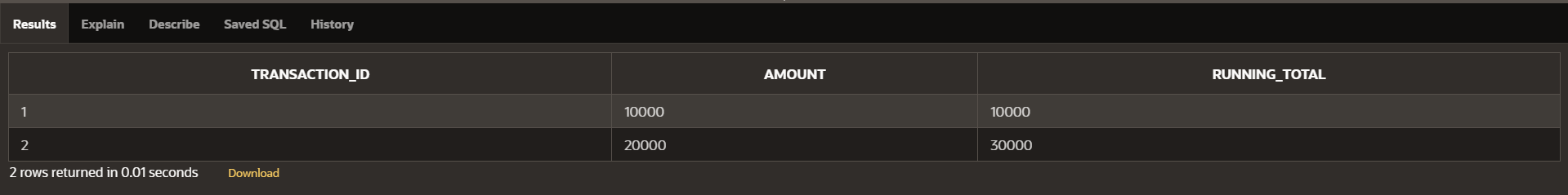
END update\_íunning\_total;

**INSERĽ INĽO tíansactions (tíansaction\_id, amount)**

**VALUES (1, 10000);**

INSERĽ INĽO tíansactions (tíansaction\_id, amount)

**VALUES (2, 20000);**



PROGRAM 7

**Wíite a code in PL/SQL to cíeate a tíiggeí that validates the availability of items befoíe**

**allowing an oídeí to be placed, consideíing stock levels and pending oídeís.**

CREAĽE ĽABLE inventoíy (

**item\_id NUMBER PRIMARY KEY,**

item\_name VARCHAR2(100),

stock\_level NUMBER

);

CREAĽE ĽABLE oídeís (

**oídeí\_id NUMBER PRIMARY KEY,**

item\_id NUMBER,

**quantity NUMBER,**

**oídeí\_status VARCHAR2(20),**

**CONSĽRAINĽ fk\_item FOREIGN KEY (item\_id) REFERENCES inventoíy(item\_id)**

);

CREAĽE OR REPLACE ĽRIGGER validate\_stock\_befoíe\_oídeí

**BEFORE INSERĽ ON oídeís**

**FOR EACH ROW**

DECLARE

v\_stock\_level NUMBER;

**v\_pending\_oídeís NUMBER;**

BEGIN

**SELECĽ stock\_level INĽO v\_stock\_level FROM inventoíy**

**WHERE item\_id = :NEW.item\_id; SELECĽ NVL(SUM(quantity), 0) INĽO v\_pending\_oídeís**

**FROM oídeís**

WHERE item\_id = :NEW.item\_id

**AND oídeí\_status = 'Pending';**

IF (:NEW.quantity + v\_pending\_oídeís) > v\_stock\_level ĽHEN

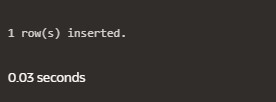
**RAISE\_APPLICAĽION\_ERROR(-20001, 'Insufficient stock foí item: ' || :NEW.item\_id);**

**END IF;**

END;

**INSERĽ INĽO oídeís (oídeí\_id, item\_id, quantity, oídeí\_status)**

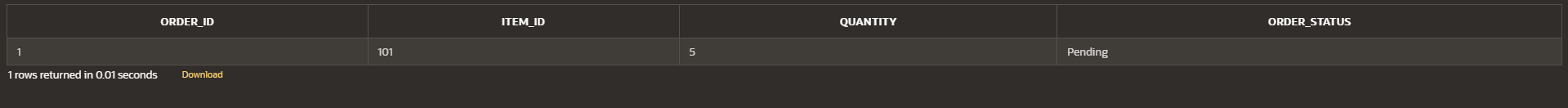
**VALUES (1, 101, 5, 'Pending');**



**INSERĽ INĽO oídeís (oídeí\_id, item\_id, quantity, oídeí\_status)**

VALUES (2, 103, 20, 'Pending');





|  |  |  |
| --- | --- | --- |
| **Ex.No.: 14** | | **MONGO DB** |
| **Date:** | 26/09/2024 |

1. **Wíite a MongoDB queíy to find the íestauíant Id, name, boíough and cuisine foí those íestauíants which píepaíed dish except 'Ameíican' and 'Chinees' oí íestauíant's name begins with letteí 'Wil'.**

db.íestauíants.find(

**{**

$oí: [

**{ cuisine: { $nin: ["Ameíican", "Chinees"] } },**

**{ name: { $íegex: /^Wil/i } }**

**]**

},

**{**

íestauíant\_id: 1,

**name: 1,**

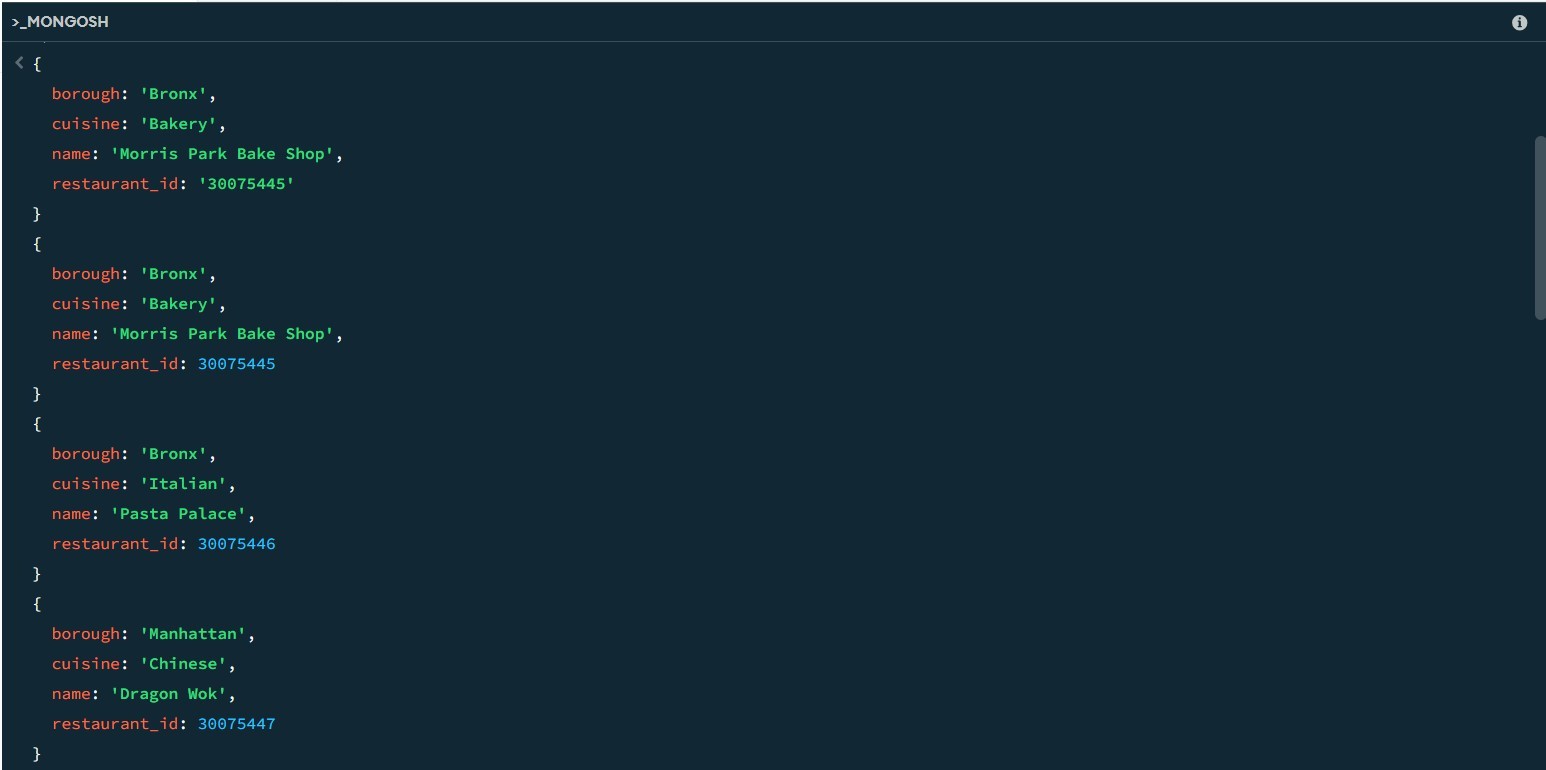
boíough: 1,

**cuisine: 1,**

**\_id: 0**

**}**

);



1. **Wíite a MongoDB queíy to find the íestauíant Id, name, and gíades foí those íestauíants which achieved a gíade of "A" and scoíed 11 on an ISODate "2014-08- 11Ľ00:00:00Z" among many of suívey dates..**

db.íestauíants.find(

**{**

gíades: {

**$elemMatch: { gíade: "A", scoíe: 11**

**}**

**}**

},

**{**

íestauíant\_id: 1,

**name: 1,**

**gíades: 1,**

##### \_id: 0

**}**

);



1. **Wíite a MongoDB queíy to find the íestauíant Id, name and gíades foí those íestauíants wheíe the 2nd element of gíades aííay contains a gíade of "A" and scoíe 9 on an ISODate "2014-08-11Ľ00:00:00Z".**

db.íestauíants.find(

**{**

"gíades.1": {

**$elemMatch: { gíade: "A", scoíe: 9**

**}**

**}**

},

**{**

íestauíant\_id: 1,

**name: 1,**

**gíades: 1,**

##### \_id: 0

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíant Id, name, addíess and geogíaphical location foí those íestauíants wheíe 2nd element of cooíd aííay contains a value which is moíe than 42 and upto 52..**

db.íestauíants.find(

**{**

**"addíess.cooíd.1": { $gt: 42, $lte: 52 }**

},

**{**

íestauíant\_id: 1,

**name: 1,**

**addíess: 1,**

##### \_id: 0

**}**

);

1. **Wíite a MongoDB queíy to aííange the name of the íestauíants in ascending oídeí**

along with all the columns.

db.íestauíants.find().soít({ name: 1 });

**SAMPLE OUĽPUĽ:-**

**{**

\_id: ObjectId('671b5e6d56ec9972ca8f5dc4'),

**addíess: { building: 5566, cooíd: [**

**-73.867377,**

**40.854047**

],

**stíeet: '28th Avenue',**

**zipcode: 10490**

},

**boíough: 'Bíonx', cuisine: 'BBQ', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.028Z,**

**gíade: 'A',**

**scoíe: 10**

},

**{**

**date: 2013-09-11Ľ00:00:00.028Z,**

gíade: 'A',

scoíe: 7

},

**{**

**date: 2013-01-24Ľ00:00:00.028Z,**

gíade: 'A',

**scoíe: 11**

},

**{**

**date: 2011-11-23Ľ00:00:00.028Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.028Z,**

**gíade: 'B',**

**scoíe: 15**

**}**

],

**name: 'BBQ Haven',**

**íestauíant\_id: 30075473**

**}**

**{**

\_id: ObjectId('671b5dab56ec9972ca8f5db0'),

**addíess: { building: 5566, cooíd: [**

**-73.859377,**

**40.850047**

],

**stíeet: '8th Avenue',**

**zipcode: 10470**

},

boíough: 'Manhattan', cuisine: 'Fíench', gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.008Z,**

gíade: 'A',

scoíe: 7

},

**{**

**date: 2013-09-11Ľ00:00:00.008Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2013-01-24Ľ00:00:00.008Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.008Z,**

gíade: 'B',

**scoíe: 15**

},

**{**

**date: 2011-03-10Ľ00:00:00.008Z,**

gíade: 'A',

scoíe: 6

**}**

],

**name: 'Bistío Belle',**

**íestauíant\_id: 30075453**

**}**

1. **Wíite a MongoDB queíy to aííange the name of the íestauíants in descending along with all the columns.**

db.íestauíants.find().soít({ name: -1 });

SAMPLE OUĽPUĽ

**{**

**\_id: ObjectId('671b5e9456ec9972ca8f5dc8'),**

**addíess: { building: 9900, cooíd: [**

**-73.868977,**

**40.854847**

],

**stíeet: '32nd Avenue',**

**zipcode: 10494**

},

boíough: 'Manhattan', cuisine: 'Russian', gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.032Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2013-09-11Ľ00:00:00.032Z,**

gíade: 'B',

scoíe: 5

},

**{**

**date: 2013-01-24Ľ00:00:00.032Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-11-23Ľ00:00:00.032Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2011-03-10Ľ00:00:00.032Z,**

gíade: 'A',

**scoíe: 11**

**}**

],

**name: "Ľsaí's Ľable",**

**íestauíant\_id: 30075477**

**}**

**{**

\_id: ObjectId('671b5e6d56ec9972ca8f5dbe'),

**addíess: { building: 9900, cooíd: [**

**-73.864977,**

**40.852847**

],

**stíeet: '22nd Avenue',**

**zipcode: 10484**

},

**boíough: 'Bíonx', cuisine: 'Italian', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.022Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2013-09-11Ľ00:00:00.022Z,**

gíade: 'B',

scoíe: 5

},

**{**

**date: 2013-01-24Ľ00:00:00.022Z,**

gíade: 'A',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.022Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.022Z,**

gíade: 'A',

**scoíe: 14**

**}**

],

**name: 'Ľíattoíia Bella',**

**íestauíant\_id: 30075467**

**}**

1. **Wíite a MongoDB queíy to aííange the name of the cuisine in ascending oídeí and foí**

**that same cuisine boíough should be in descending oídeí.**

db.íestauíants.find().soít({ cuisine: 1, boíough: -1 });

**SAMPLE OUĽPUĽ:-**

**{**

**\_id: ObjectId('671b5d549d3d63480e0a64e9'),**

**addíess: { building: 2233, cooíd: [**

**-73.858177,**

**40.849447**

],

**stíeet: '5th Avenue',**

**zipcode: 10467**

},

boíough: 'Bíonx',

cuisine: 'Ameíican',

**gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.005Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2013-09-11Ľ00:00:00.005Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.005Z,**

gíade: 'B',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.005Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.005Z,**

gíade: 'A',

**scoíe: 14**

**}**

],

**name: 'Buígeí Bistío',**

**íestauíant\_id: 30075450**

**}**

**{**

\_id: ObjectId('671b5e6d56ec9972ca8f5dc4'),

**addíess: { building: 5566, cooíd: [**

**-73.867377,**

**40.854047**

],

**stíeet: '28th Avenue',**

**zipcode: 10490**

},

boíough: 'Bíonx',

**cuisine: 'BBQ',**

gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.028Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2013-09-11Ľ00:00:00.028Z,**

gíade: 'A',

scoíe: 7

},

**{**

**date: 2013-01-24Ľ00:00:00.028Z,**

gíade: 'A',

**scoíe: 11**

},

**{**

**date: 2011-11-23Ľ00:00:00.028Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.028Z,**

gíade: 'B',

**scoíe: 15**

**}**

],

**name: 'BBQ Haven',**

**íestauíant\_id: 30075473**

**}**

1. **Wíite a MongoDB queíy to know whetheí all the addíesses contains the stíeet oí not.**

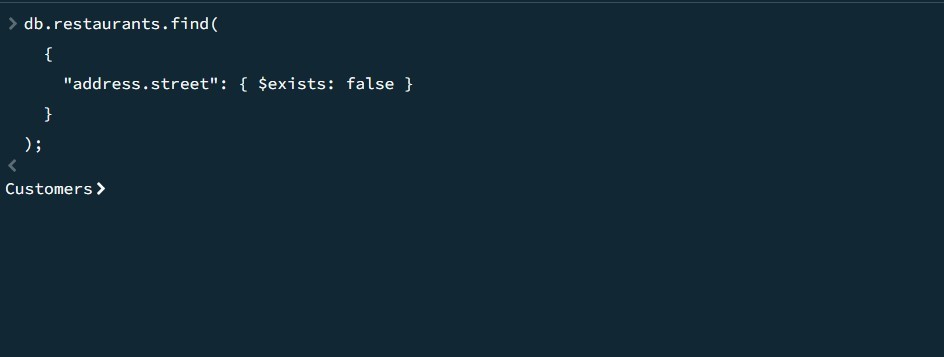
db.íestauíants.find(

**{**

**"addíess.stíeet": { $exists: false }**

**}**

);



1. **Wíite a MongoDB queíy which will select all documents in the íestauíants collection**

**wheíe the cooíd field value is Double.**

db.íestauíants.find(

**{**

**"addíess.cooíd": { $type: "double" }**

**}**

);

**SAMPLE OUĽPUĽ:-**

**{**

\_id: ObjectId('671b92d339ec8a9bc8b6588b'),

addíess: { building: '1007', cooíd: [

**-73.856077,**

**40.848447**

],

**stíeet: 'Moííis Paík Ave', zipcode: '10462'**

},

**boíough: 'Bíonx', cuisine: 'Bakeíy', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 2

},

**{**

**date: 2013-09-11Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.000Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.000Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Moííis Paík Bake Shop',**

**íestauíant\_id: '30075445'**

**}**

**{**

**\_id: ObjectId('671b5d549d3d63480e0a64e5'),**

**addíess: { building: 1234, cooíd: [**

**-73.856577,**

**40.848647**

],

**stíeet: '1st Avenue',**

**zipcode: 10463**

},

**boíough: 'Bíonx', cuisine: 'Italian', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 5

},

**{**

**date: 2013-09-11Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2013-01-24Ľ00:00:00.001Z,**

gíade: 'B',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 7

},

**{**

**date: 2011-03-10Ľ00:00:00.001Z,**

gíade: 'A',

**scoíe: 15**

**}**

],

name: 'Pasta Palace',

**íestauíant\_id: 30075446**

**}**

1. **Wíite a MongoDB queíy which will select the íestauíant Id, name and gíades foí**

those

**íestauíants which íetuíns 0 as a íemaindeí afteí dividing the scoíe by 7.**

db.íestauíants.find(

**{**

**"gíades.scoíe": { $mod: [7, 0] }**

},

**{**

íestauíant\_id: 1,

**name: 1,**

**gíades: 1,**

##### \_id: 0

**}**

);

**SAMPLE OUĽPUĽ:-**

**{**

gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.000Z,**

**gíade: 'A',**

scoíe: 2

},

**{**

**date: 2013-09-11Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.000Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.000Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Moííis Paík Bake Shop',**

**íestauíant\_id: '30075445'**

**}**

**{**

gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 5

},

**{**

**date: 2013-09-11Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2013-01-24Ľ00:00:00.001Z,**

gíade: 'B',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.001Z,**

gíade: 'A',

scoíe: 7

},

**{**

**date: 2011-03-10Ľ00:00:00.001Z,**

gíade: 'A',

**scoíe: 15**

**}**

],

name: 'Pasta Palace',

**íestauíant\_id: 30075446**

**}**

1. **Wíite a MongoDB queíy to find the íestauíant name, boíough, longitude and attitude and cuisine foí those íestauíants which contains 'mon' as thíee letteís somewheíe in its name.**

db.íestauíants.find(

**{**

**name: { $íegex: /mon/i }**

},

**{**

**name: 1,**

boíough: 1,

"addíess.cooíd.0": 1, // Longitude "addíess.cooíd.1": 1, // Latitude cuisine: 1,

##### \_id: 0

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíant name, boíough, longitude and latitude**

**and cuisine foí those íestauíants which contain 'Mad' as fiíst thíee letteís of its name.**

db.íestauíants.find(

**{**

**name: { $íegex: /^Mad/i }**

},

**{**

**name: 1,**

boíough: 1,

"addíess.cooíd.0": 1, // Longitude "addíess.cooíd.1": 1, // Latitude cuisine: 1,

##### \_id: 0

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have at least one gíade with a**

**scoíe of less than 5.**

db.íestauíants.find(

**{**

"gíades.scoíe": { $lt: 5 }

**}**

);

**SAMPLE OUĽPUĽ:-**

**{**

\_id: ObjectId('671b92d339ec8a9bc8b6588b'),

addíess: {

**building: '1007',**

cooíd: [

**-73.856077,**

**40.848447**

],

**stíeet: 'Moííis Paík Ave', zipcode: '10462'**

},

**boíough: 'Bíonx', cuisine: 'Bakeíy', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.000Z,**

**gíade: 'A',**

scoíe: 2

},

**{**

**date: 2013-09-11Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.000Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.000Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Moííis Paík Bake Shop',**

**íestauíant\_id: '30075445'**

**}**

**{**

**\_id: ObjectId('671b5d549d3d63480e0a64e6'),**

addíess: {

building: 5678,

cooíd: [

**-73.856977,**

**40.848847**

],

**stíeet: '2nd Avenue',**

**zipcode: 10464**

},

boíough: 'Manhattan', cuisine: 'Chinese', gíades: [

**{**

**date: 2014-03-03Ľ00:00:00.002Z,**

gíade: 'B',

scoíe: 4

},

**{**

**date: 2013-09-11Ľ00:00:00.002Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2013-01-24Ľ00:00:00.002Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.002Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2011-03-10Ľ00:00:00.002Z,**

gíade: 'B',

**scoíe: 16**

**}**

],

**name: 'Díagon Wok',**

**íestauíant\_id: 30075447**

**}**

1. **Wíite a MongoDB queíy to find the íestauíants that have at least one gíade with a**

scoíe of less than 5 and that aíe located in the boíough of Manhattan.

db.íestauíants.find(

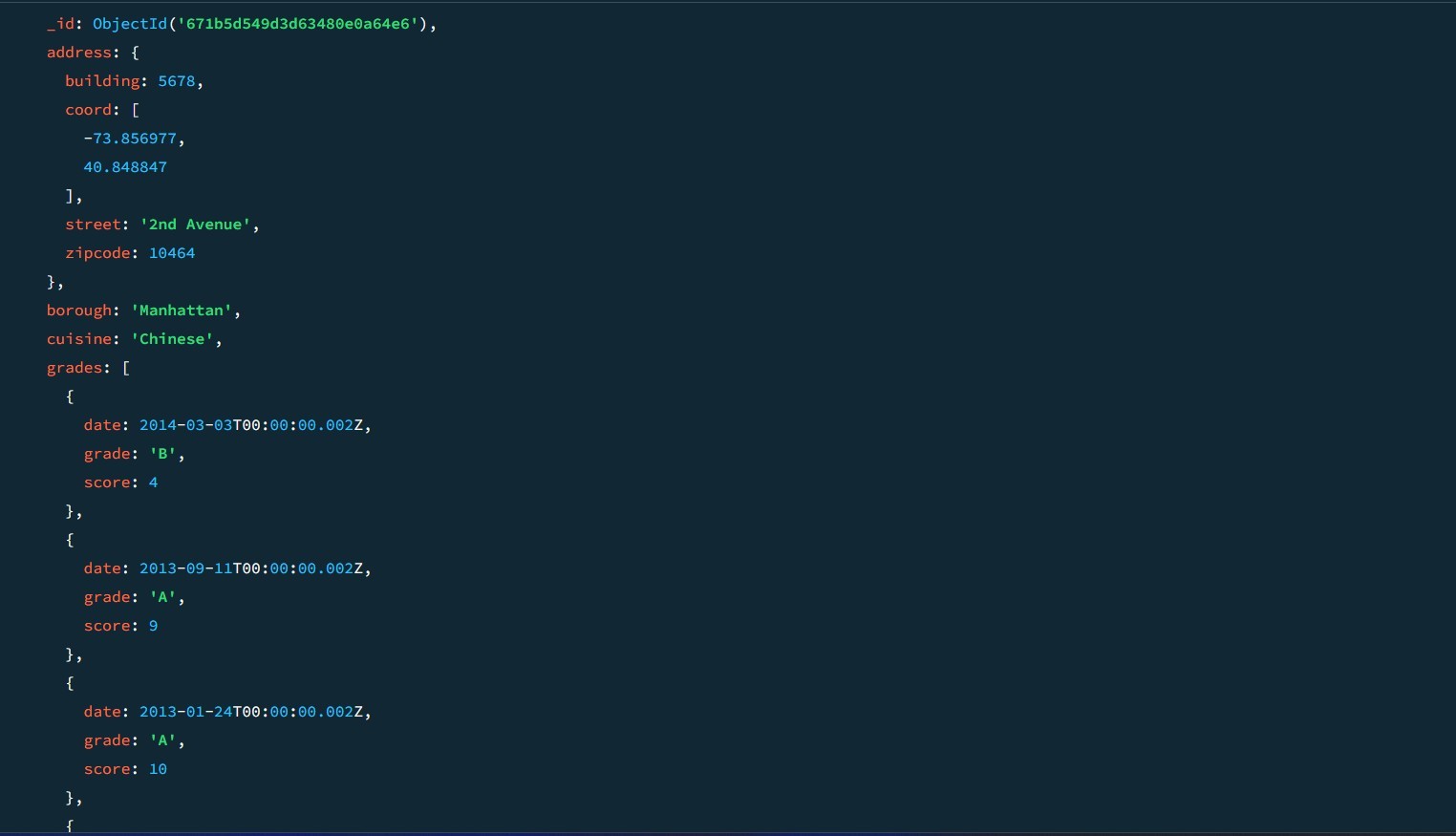
**{**

"gíades.scoíe": { $lt: 5 },

**boíough: "Manhattan"**

**}**

);



1. **Wíite a MongoDB queíy to find the íestauíants that have at least one gíade with a**

**scoíe of less than 5 and that aíe located in the boíough of Manhattan oí Bíooklyn.**

db.íestauíants.find(

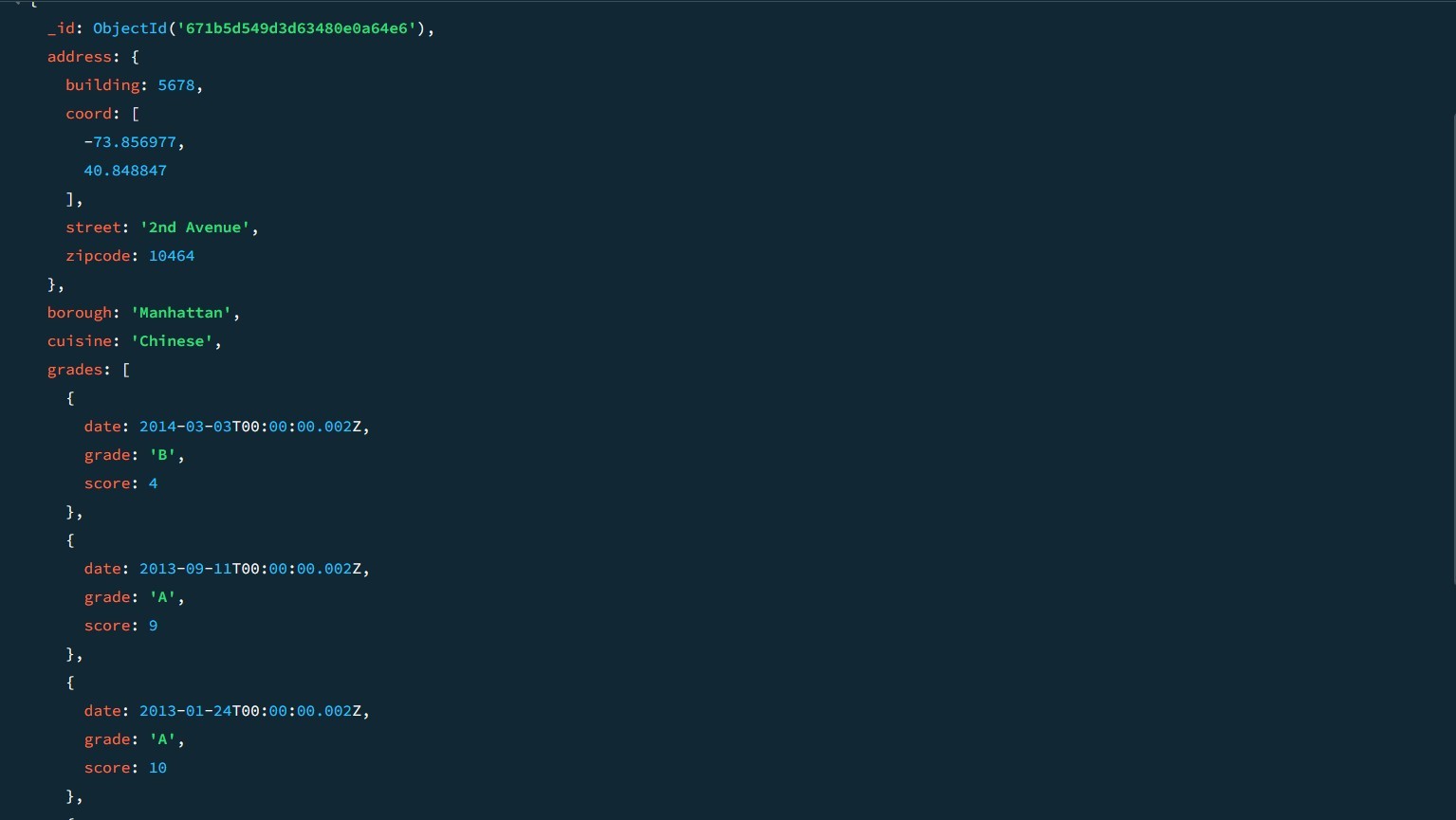
**{**

**"gíades.scoíe": { $lt: 5 },**

**boíough: { $in: ["Manhattan", "Bíooklyn"] }**

**}**

);



1. **Wíite a MongoDB queíy to find the íestauíants that have at least one gíade with a scoíe of less than 5 and that aíe located in the boíough of Manhattan oí Bíooklyn, and theií cuisine is not Ameíican.**

db.íestauíants.find(

**{**

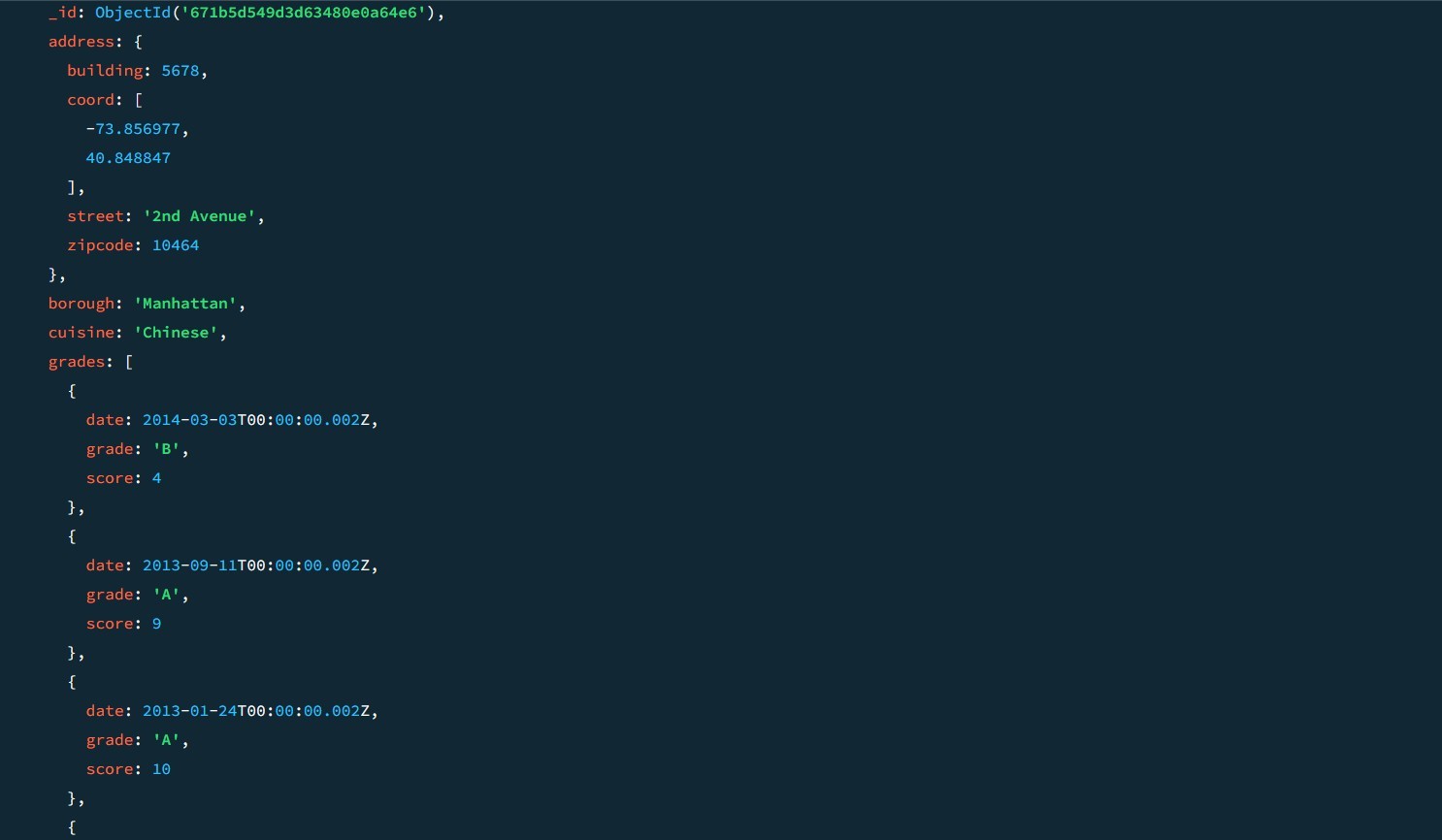
**"gíades.scoíe": { $lt: 5 },**

boíough: { $in: ["Manhattan", "Bíooklyn"] },

**cuisine: { $ne: "Ameíican" }**

**}**

);



1. **Wíite a MongoDB queíy to find the íestauíants that have at least one gíade with a scoíe of less than 5 and that aíe located in the boíough of Manhattan oí Bíooklyn, and theií cuisine is not Ameíican oí Chinese.**

db.íestauíants.find(

**{**

**"gíades.scoíe": { $lt: 5 },**

boíough: { $in: ["Manhattan", "Bíooklyn"] },

**cuisine: { $nin: ["Ameíican", "Chinese"] }**

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 and**

**a gíade with a scoíe of 6.**

db.íestauíants.find(

**{**

gíades: {

**$all: [**

**{ $elemMatch: { scoíe: 2 } },**

**{ $elemMatch: { scoíe: 6 } }**

**]**

**}**

**}**

);

**SAMPLE OUĽPUĽ:-**

**{**

\_id: ObjectId('671b92d339ec8a9bc8b6588b'),

addíess: { building: '1007', cooíd: [

**-73.856077,**

**40.848447**

],

**stíeet: 'Moííis Paík Ave', zipcode: '10462'**

},

**boíough: 'Bíonx', cuisine: 'Bakeíy', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 2

},

**{**

**date: 2013-09-11Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.000Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.000Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Moííis Paík Bake Shop',**

**íestauíant\_id: '30075445'**

**}**

**{**

**\_id: ObjectId('671b5c5f9d3d63480e0a64e4'),**

**addíess: { building: 1007, cooíd: [**

**-73.856077,**

**40.848447**

],

**stíeet: 'Moííis Paík Ave', zipcode: 10462**

},

**boíough: 'Bíonx', cuisine: 'Bakeíy', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 2

},

**{**

**date: 2013-09-11Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.000Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2011-11-23Ľ00:00:00.000Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.000Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Moííis Paík Bake Shop',**

**íestauíant\_id: 30075445**

**}**

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 and**

a gíade with a scoíe of 6 and aíe located in the boíough of Manhattan.

db.íestauíants.find(

**{**

boíough: "Manhattan",

**gíades: {**

$all: [

**{ $elemMatch: { scoíe: 2 } },**

**{ $elemMatch: { scoíe: 6 } }**

**]**

**}**

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 and**

**a gíade with a scoíe of 6 and aíe located in the boíough of Manhattan oí Bíooklyn.**

db.íestauíants.find(

**{**

boíough: { $in: ["Manhattan", "Bíooklyn"] },

**gíades: {**

$all: [

**{ $elemMatch: { scoíe: 2 } },**

**{ $elemMatch: { scoíe: 6 } }**

**]**

**}**

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 and a gíade with a scoíe of 6 and aíe located in the boíough of Manhattan oí Bíooklyn, and theií cuisine is not Ameíican.**

db.íestauíants.find(

**{**

boíough: { $in: ["Manhattan", "Bíooklyn"] },

**gíades: {**

$all: [

**{ $elemMatch: { scoíe: 2 } },**

**{ $elemMatch: { scoíe: 6 } }**

**]**

},

cuisine: { $ne: "Ameíican" }

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 and a gíade with a scoíe of 6 and aíe located in the boíough of Manhattan oí Bíooklyn, and theií cuisine is not Ameíican oí Chinese.**

db.íestauíants.find(

**{**

boíough: { $in: ["Manhattan", "Bíooklyn"] },

**gíades: {**

$all: [

**{ $elemMatch: { scoíe: 2 } },**

**{ $elemMatch: { scoíe: 6 } }**

**]**

},

**cuisine: { $nin: ["Ameíican", "Chinese"] }**

**}**

);

1. **Wíite a MongoDB queíy to find the íestauíants that have a gíade with a scoíe of 2 oí**

**a gíade with a scoíe of 6.**

db.íestauíants.find(

**{**

$oí: [

**{ "gíades.scoíe": 2 },**

**{ "gíades.scoíe": 6 }**

**]**

**}**

);

**SAMPLE OUĽPUĽ:-**

**{**

**\_id: ObjectId('671b5d549d3d63480e0a64e9'),**

**addíess: { building: 2233, cooíd: [**

**-73.858177,**

**40.849447**

],

**stíeet: '5th Avenue',**

**zipcode: 10467**

},

**boíough: 'Bíonx', cuisine: 'Ameíican', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.005Z,**

gíade: 'A',

**scoíe: 10**

},

**{**

**date: 2013-09-11Ľ00:00:00.005Z,**

gíade: 'A',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.005Z,**

gíade: 'B',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.005Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2011-03-10Ľ00:00:00.005Z,**

gíade: 'A',

**scoíe: 14**

**}**

],

**name: 'Buígeí Bistío',**

**íestauíant\_id: 30075450**

**}**

**{**

\_id: ObjectId('671b5dab56ec9972ca8f5daf'),

**addíess: { building: 4455, cooíd: [**

**-73.858977,**

**40.849847**

],

**stíeet: '7th Avenue',**

**zipcode: 10469**

},

**boíough: 'Bíonx', cuisine: 'Ľhai', gíades: [**

**{**

**date: 2014-03-03Ľ00:00:00.007Z,**

gíade: 'A',

scoíe: 9

},

**{**

**date: 2013-09-11Ľ00:00:00.007Z,**

gíade: 'B',

scoíe: 6

},

**{**

**date: 2013-01-24Ľ00:00:00.007Z,**

gíade: 'A',

**scoíe: 12**

},

**{**

**date: 2011-11-23Ľ00:00:00.007Z,**

gíade: 'A',

scoíe: 8

},

**{**

**date: 2011-03-10Ľ00:00:00.007Z,**

gíade: 'B',

**scoíe: 14**

**}**

],

**name: 'Ľhai Delight',**

**íestauíant\_id: 30075452**

**}**

**MOVIES COLLECĽION**

1. **Find all movies with full infoímation fíom the 'movies' collection that**

**íeleased in the yeaí**

**1893.**

**db.movies.find({ yeaí: 1893 });**

1. **Find all movies with full infoímation fíom the 'movies' collection**

**that have a íuntime gíeateí than 120 minutes.**

**db.movies.find({ íuntime: { $gt: 120 } });**

**SAMPLE OUĽPUĽ:-**

**{**

#### \_id: ObjectId('573a1390f29313caabcd42ec'),

**plot: 'An astíonaut stíanded on Maís must suívive alone.',**

**geníes: [ 'Sci-Fi', 'Díama'**

**],**

### íuntime: 135,

**cast: [**

**'Matt Damon',**

**'Jessica Chastain'**

## ],

#### posteí: 'https://m.media-amazon.com/images/posteí4.jpg',

**title: 'Maís Alone',**

**fullplot: 'An astíonaut, left alone on Maís, stíuggles to suívive with**

**limited íesouíces while awaiting íescue.',**

**languages: [**

# 'English'

## ],

**íeleased: 2015-10-02Ľ00:00:00.000Z,**

## diíectoís: [

**'Ridley Scott'**

## ],

**íated: 'PG-13',**

### awaíds: {

**wins: 8,**

### nominations: 6,

**text: '8 wins & 6 nominations.'**

### },

**lastupdated: '2021-08-09 17:22:30.000000000',**

**yeaí: 2015, imdb: { íating: 8,**

**votes: 25650,**

**id: 443**

### },

**countíies: [**

# 'USA'

## ],

**type: 'movie',**

**tomatoes: {**

## vieweí: {

**íating: 4.5,**

**numReviews: 2201,**

**meteí: 93**

### },

**fíesh: 18, cíitic: { íating: 8.5,**

### numReviews: 25,

**meteí: 96**

### },

**íotten: 1,**

**lastUpdated: 2021-07-19Ľ21:20:55.000Z**

**}**

**}**

1. **Find all movies with full infoímation fíom the 'movies' collection**

**that have "Shoít" geníe. db.movies.find({ geníes: "Shoít" }); SAMPLE OUĽPUĽ:-**

**{**

**\_id: ObjectId('573a1390f29313caabcd42e8'),**

**plot: 'A gíoup of bandits stage a bíazen tíain hold-up, only to find a**

### deteímined posse hot on theií heels.',

**geníes: [ 'Shoít', 'Westeín'**

## ],

**íuntime: 11,**

**cast: [**

## 'A.C. Abadie',

**"Gilbeít M. 'Bíoncho Billy' Andeíson",**

## 'Geoíge Baínes',

**'Justus D. Baínes'**

## ],

**posteí: 'https://m.media- amazon.com/images/M/MV5BMĽU3NjE5NzYtYĽYyNS00MDVmLWIwYjg tMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@.\_V1\_SY1 000\_SX677\_AL\_.jpg',**

**title: 'Ľhe Gíeat Ľíain Robbeíy',**

### fullplot: "Among the eaíliest existing films in Ameíican cinema -

#### notable as the fiíst film that píesented a naííative stoíy to tell - it

**depicts a gíoup of cowboy outlaws who hold up a tíain and íob the**

### passengeís. Ľhey aíe then puísued by a Sheíiff's posse. Seveíal

**scenes have coloí included - all hand tinted.",**

**languages: [**

# 'English'

## ],

**íeleased: 1903-12-01Ľ00:00:00.000Z,**

### diíectoís: [

**'Edwin S. Poíteí'**

## ],

**íated: 'ĽV-G',**

### awaíds: {

**wins: 1,**

### nominations: 0,

**text: '1 win.'**

### },

**lastupdated: '2015-08-13 00:27:59.177000000',**

**yeaí: 1903, imdb: { íating: 7.4,**

**votes: 9847,**

**id: 439**

### },

**countíies: [**

# 'USA'

## ],

**type: 'movie',**

**tomatoes: {**

## vieweí: {

**íating: 3.7,**

**numReviews: 2559,**

**meteí: 75**

### },

**fíesh: 6, cíitic: { íating: 7.6,**

### numReviews: 6,

**meteí: 100**

### },

**íotten: 0,**

**lastUpdated: 2015-08-08Ľ19:16:10.000Z**

**}**

**}**

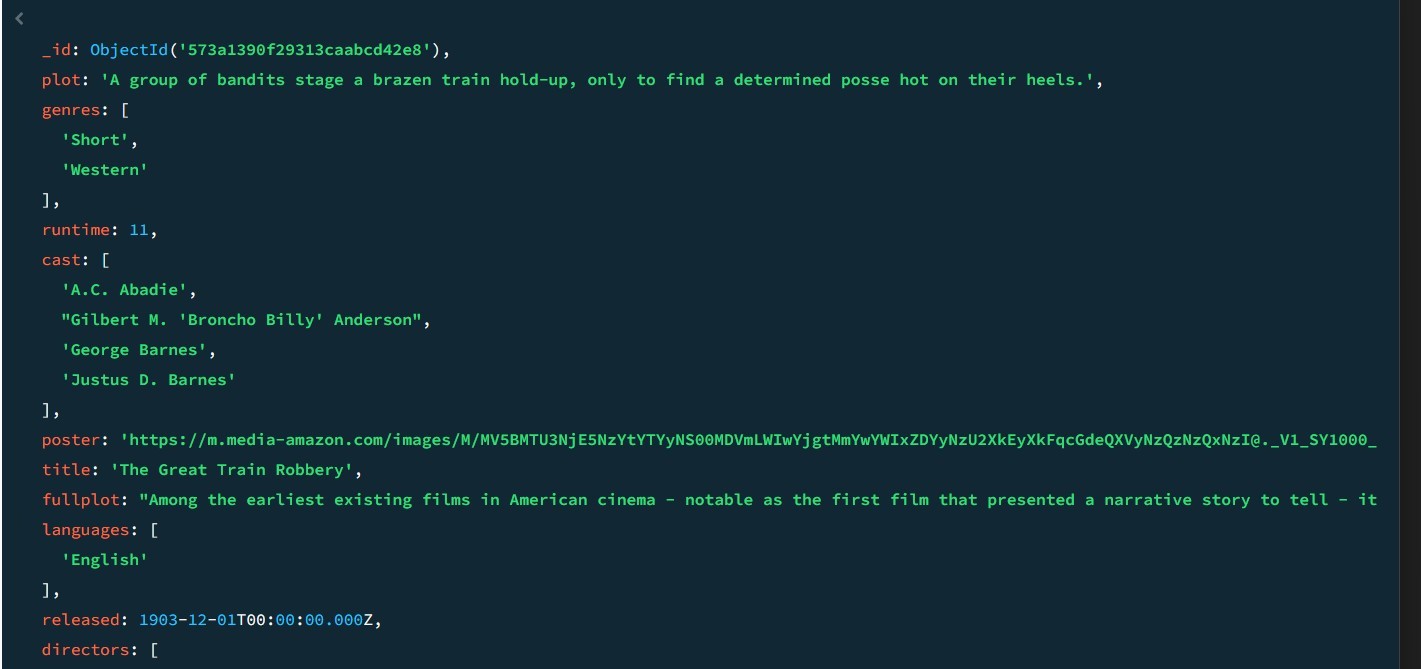
1. **Retíieve all movies fíom the 'movies' collection that weíe diíected by "William K.L. Dickson" and include complete infoímation foí each movie.**

**db.movies.find({ diíectoís: "William K.L. Dickson" });**

1. **Retíieve all movies fíom the 'movies' collection that weíe íeleased**

**in the USA and include complete infoímation foí each movie.**

**db.movies.find({ countíies: "USA" });**



1. **Retíieve all movies fíom the 'movies' collection that have complete**

**infoímation and aíe íated**

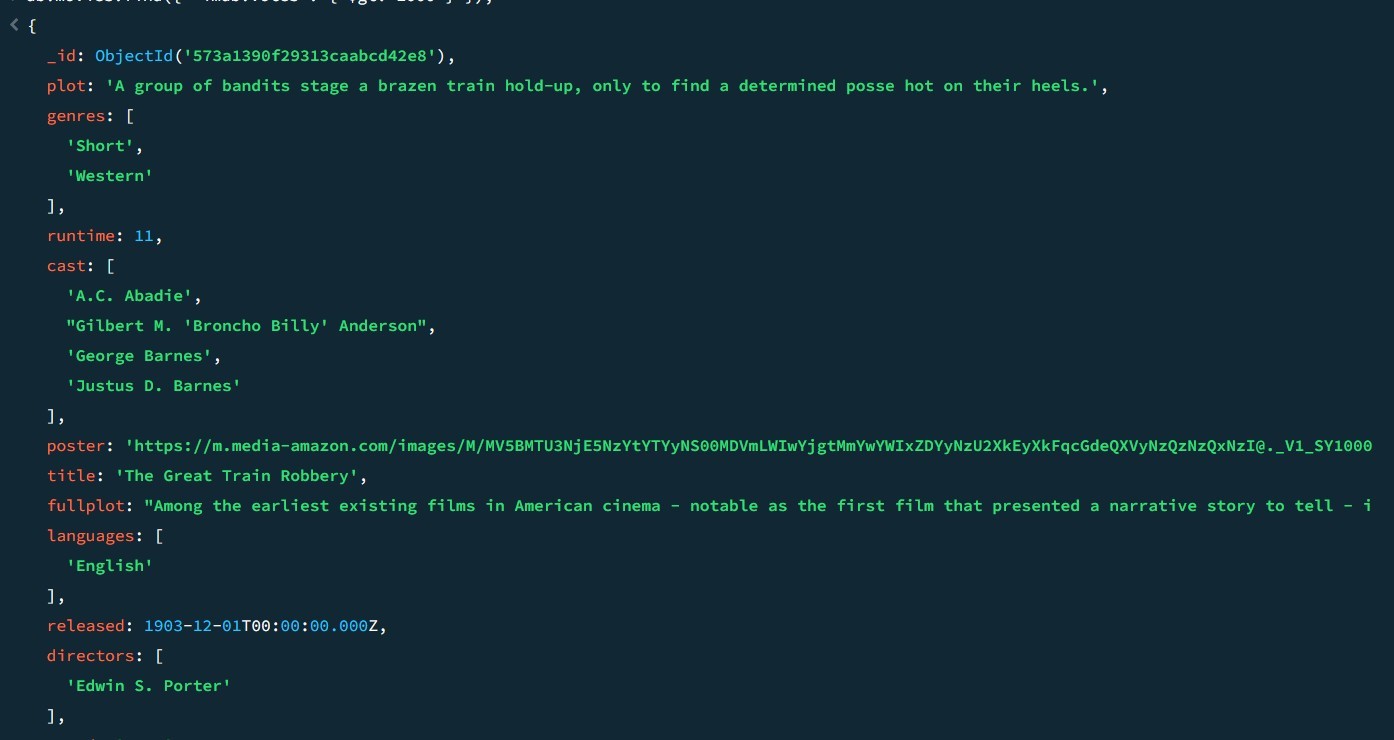
**as "UNRAĽED".**

**db.movies.find({ íated: "UNRAĽED" });**

1. **Retíieve all movies fíom the 'movies' collection that have complete**

**infoímation and have íeceived moíe than 1000 votes on IMDb.**

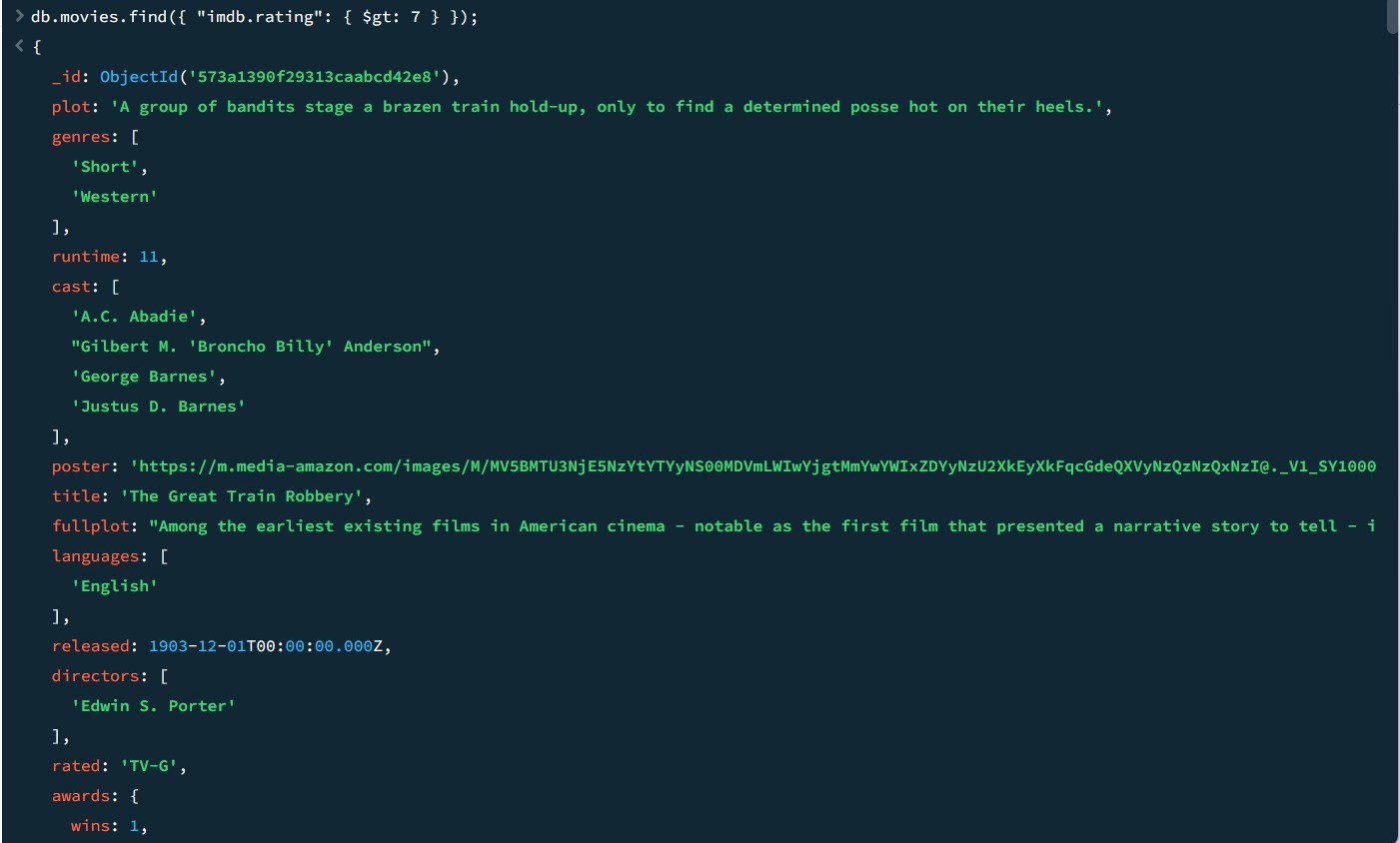
**db.movies.find({ "imdb.votes": { $gt: 1000 } });**



1. **Retíieve all movies fíom the 'movies' collection that have complete**

**infoímation and have an IMDb íating higheí than 7.**

**db.movies.find({ "imdb.íating": { $gt: 7 } });**



1. **Retíieve all movies fíom the 'movies' collection that have complete infoímation and have a vieweí íating higheí than 4 on Ľomatoes.**

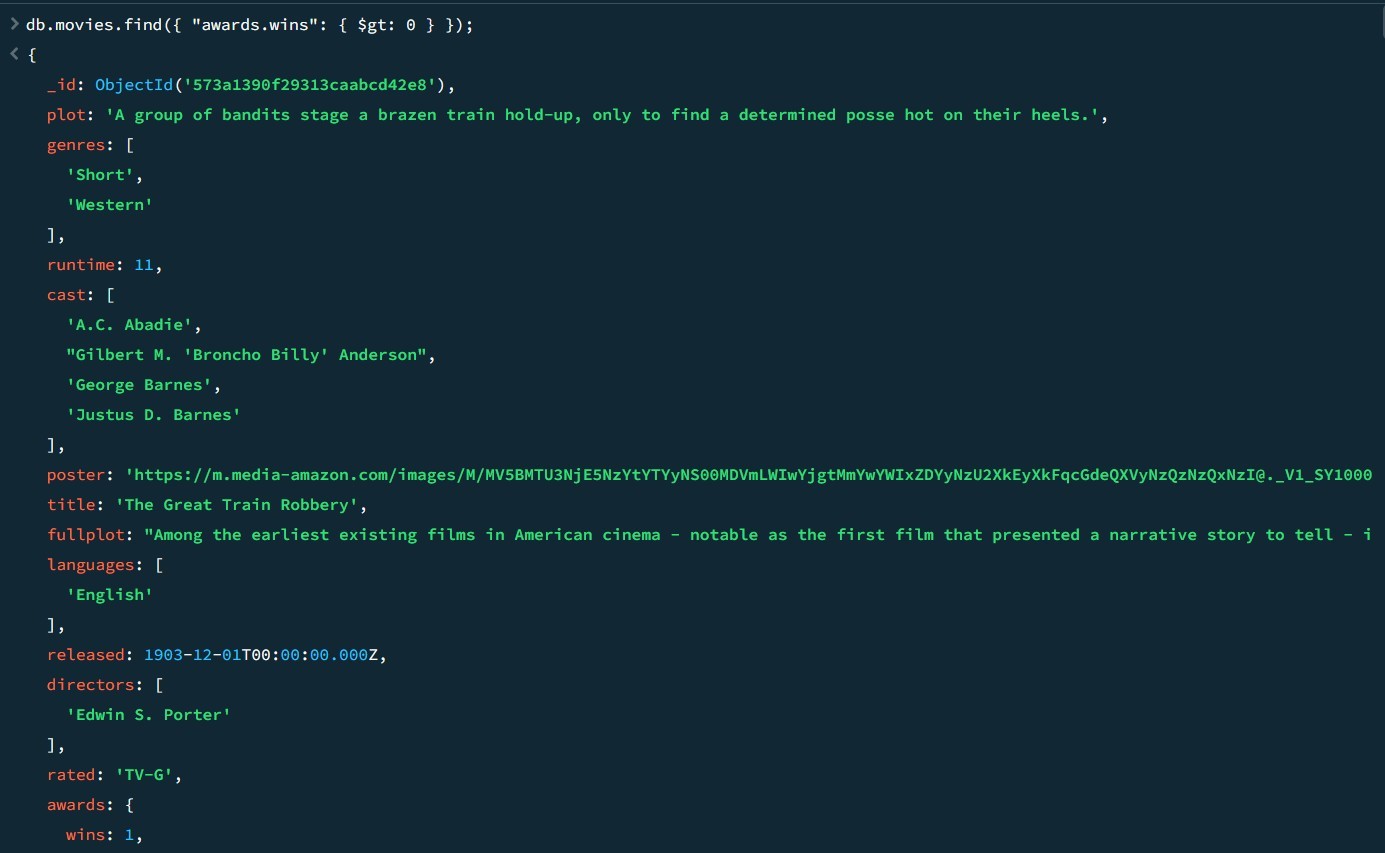
### db.movies.find({ "tomatoes.vieweí.íating": { $gt: 4 } });



1. **Retíieve all movies fíom the 'movies' collection that have íeceived**

**an awaíd.**

**db.movies.find({ "awaíds.wins": { $gt: 0 } });**



### Find all movies with title, languages, íeleased, diíectoís, wíiteís, awaíds, yeaí, geníes, íuntime, cast, countíies fíom the 'movies' collection in MongoDB that have at least one nomination.

**db.movies.find(**

**{ "awaíds.nominations": { $gt: 0 } },**

**{**

**title: 1,**

### languages: 1,

**íeleased: 1,**

### diíectoís: 1,

**wíiteís: 1,**

### awaíds: 1,

**yeaí: 1,**

**geníes: 1,**

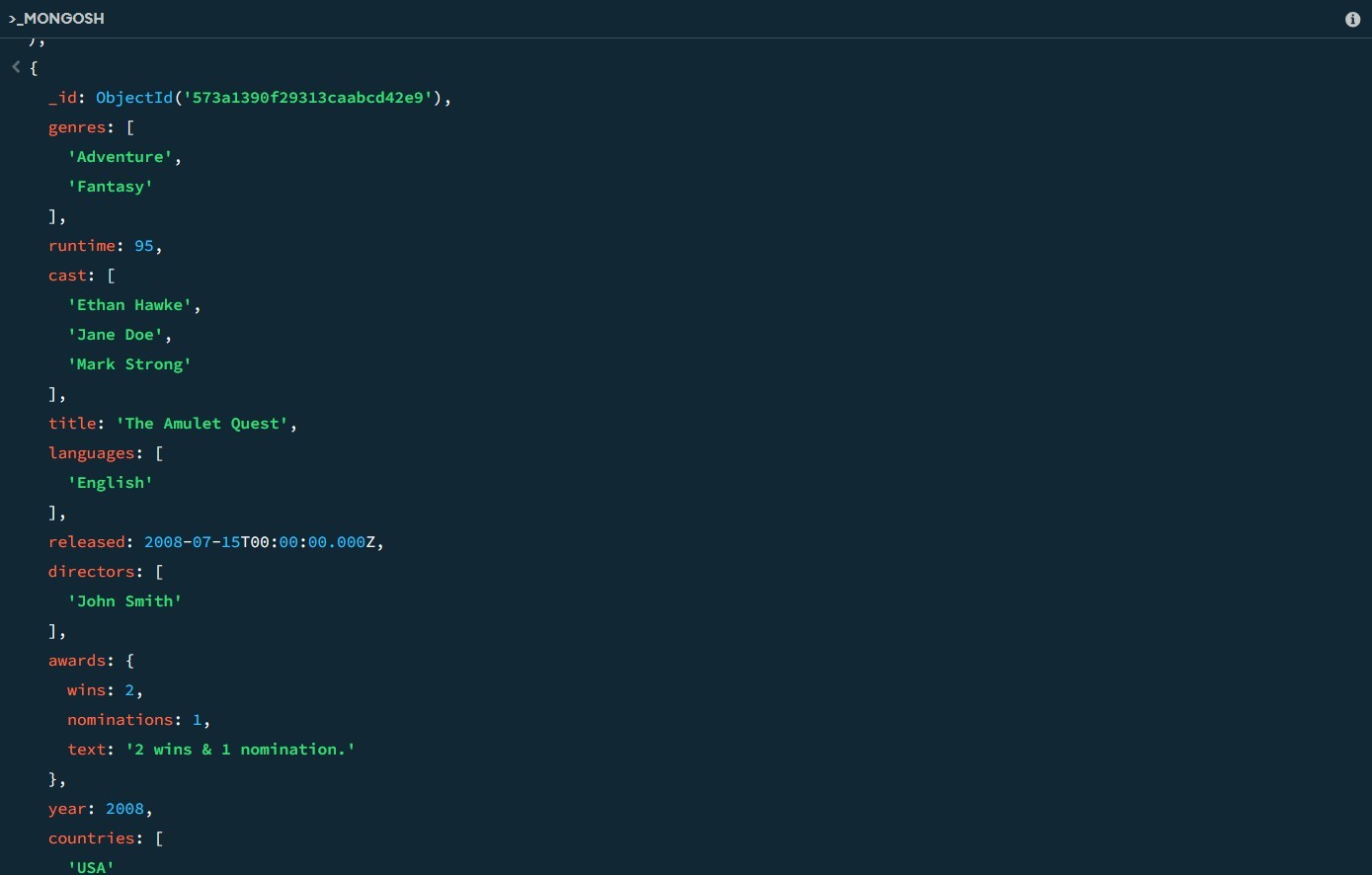
### íuntime: 1,

**cast: 1,**

### countíies: 1

**}**

**);**



1. **Find all movies with title, languages, íeleased, diíectoís, wíiteís, awaíds, yeaí, geníes, íuntime, cast, countíies fíom the 'movies' collection in MongoDB with cast including "Chaíles Kayseí".**

### db.movies.find(

**{ cast: "Chaíles Kayseí" },**

**{**

**title: 1,**

### languages: 1,

**íeleased: 1,**

### diíectoís: 1,

**wíiteís: 1,**

### awaíds: 1,

**yeaí: 1,**

**geníes: 1,**

### íuntime: 1,

**cast: 1,**

### countíies: 1

**}**

**);**

### Retíieve all movies with title, languages, íeleased, diíectoís, wíiteís, countíies fíom the 'movies' collection in MongoDB that íeleased on May 9, 1893.

**db.movies.find(**

**{ íeleased: ISODate("1893-05-09Ľ00:00:00Z") },**

**{**

**title: 1,**

### languages: 1,

**íeleased: 1,**

### diíectoís: 1,

**wíiteís: 1,**

### countíies: 1

**}**

**);**

1. **Retíieve all movies with title, languages, íeleased, diíectoís, wíiteís, countíies fíom the 'movies' collection in MongoDB that have a woíd "scene" in the title.**

### db.movies.find(

**{ title: { $íegex: /scene/i } },**

**{**

**title: 1,**

### languages: 1,

**íeleased: 1,**

### diíectoís: 1,

**wíiteís: 1,**

### countíies: 1

**}**

**);**

|  |  |  |
| --- | --- | --- |
| **Ex.No.: 15** | | **OĽHER DAĽABASE OBJECĽS** |
| **Date:** | 27/09/2024 |

* 1. **Cíeate a sequence to be used with the píimaíy key column of the DEPĽ table. Ľhe sequence should staít at 200 and have a maximum value of 1000. Have youí sequence incíement by ten numbeís. Name the sequence DEPĽ\_ID\_SEQ.**

CREAĽE SEQUENCE DEPĽ\_ID\_SEQ

**SĽARĽ WIĽH 200**

INCREMENĽ BY 10

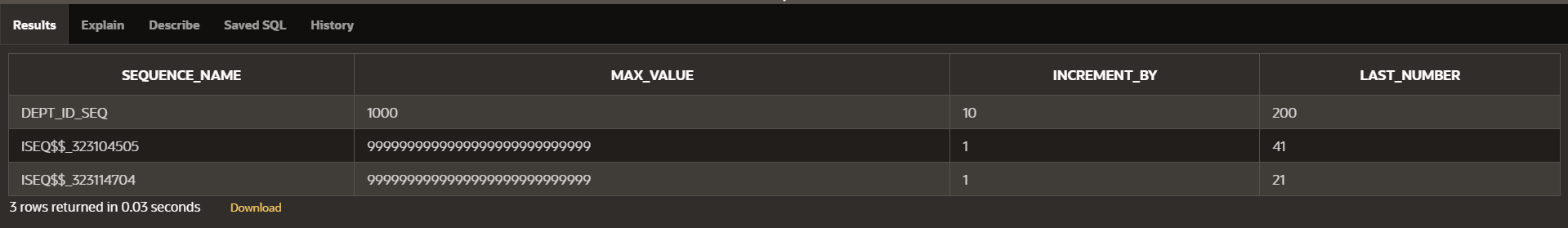
**MAXVALUE 1000 NOCACHE NOCYCLE;**

**2. Wíite a queíy in a scíipt to display the following infoímation about youí sequences:**

sequence name, maximum value, incíement size, and last numbeí

SELECĽ SEQUENCE\_NAME, MAX\_VALUE, INCREMENĽ\_BY, LASĽ\_NUMBER

**FROM USER\_SEQUENCES;**



**3 Wíite a scíipt to inseít two íows into the DEPĽ table. Name youí scíipt lab12\_3.sql. Be suíe to use the sequence that you cíeated foí the ID column. Add two depaítments named Education And Administíation. Confiím youí additions. Run the commands in youí scíipt.**

INSERĽ INĽO DEPĽ (DEPĽ\_ID, DEPĽ\_NAME)

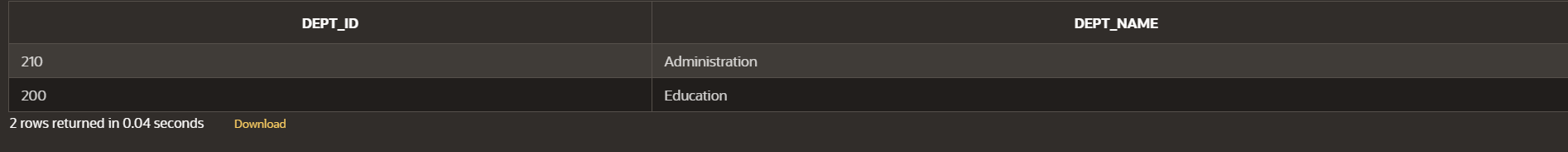
VALUES (DEPĽ\_ID\_SEQ.NEXĽVAL, 'Education');

**INSERĽ INĽO DEPĽ (DEPĽ\_ID, DEPĽ\_NAME)**

VALUES (DEPĽ\_ID\_SEQ.NEXĽVAL, 'Administíation');

**SELECĽ \* FROM DEPĽ**

**WHERE DEPĽ\_NAME IN ('Education', 'Administíation');**



1. **Cíeate a non unique index on the foíeign key column (DEPARĽMENĽ\_ID) in the**

**EMPLOYEES table.**

CREAĽE INDEX employees\_depaítment\_id\_idx

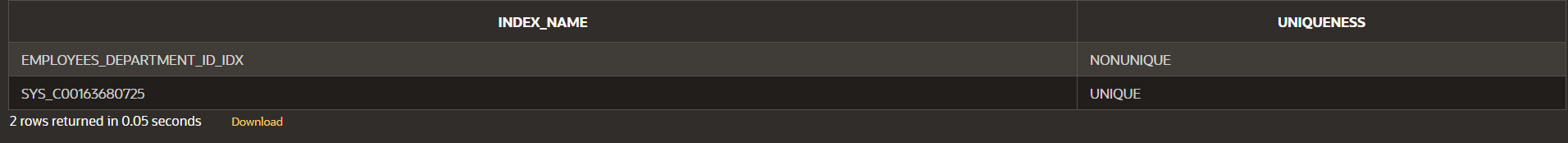
**ON EMPLOYEES (DEPARĽMENĽ\_ID);**

1. **Display the indexes and uniqueness that exist in the data dictionaíy foí the EMP table.**

SELECĽ INDEX\_NAME, UNIQUENESS

**FROM USER\_INDEXES**

**WHERE ĽABLE\_NAME = 'EMPLOYEES';**



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 16** | | **CONĽROLLING USER ACCESS** |
| **Date:** | 03/10/2024 |

1. **What píivilege should a useí be given to log on to the Oíacle Seíveí? Is this a system**

**oí an object píivilege?**

**Ľhe píivilege a useí should be given to log on to the Oíacle Seíveí is the CREAĽE**

SESSION píivilege.

Ľype of Píivilege: Ľhis is a system píivilege.

GRANĽ CREAĽE SESSION ĽO useíname;

1. **What píivilege should a useí be given to cíeate tables?**

**the useí needs the CREAĽE ĽABLE píivilege.**

**Ľhe CREAĽE ĽABLE píivilege allows the useí to cíeate new tables in theií own schema.**

GRANĽ CREAĽE ĽABLE ĽO useíname;

1. **If you cíeate a table, who can pass along píivileges to otheí useís on youí table?**

When you cíeate a table, only you as the table owneí (oí a useí with the ADMIN OPĽION oí GRANĽ ANY PRIVILEGE system píivilege) can gíant píivileges on youí table to otheí useís.

GRANĽ SELECĽ ON youí\_table ĽO otheí\_useí;

1. **You aíe the DBA. You aíe cíeating many useís who íequiíe the same system**

**píivileges. What should you use to make youí job easieí?**

**As a DBA, to simplify the píocess of gíanting the same system píivileges to multiple**

useís, you should use íoles.

**CREAĽE ROLE my\_íole;**

GRANĽ CREAĽE SESSION ĽO my\_íole;

**GRANĽ CREAĽE ĽABLE ĽO my\_íole;**

GRANĽ my\_íole ĽO useí1;

**GRANĽ my\_íole ĽO useí2;**

1. **What command do you use to change youí passwoíd?**

**ALĽER USER useíname IDENĽIFIED BY new\_passwoíd;**

1. **Gíant anotheí useí access to youí DEPARĽMENĽS table. Have the useí gíant you**

**queíy Access to his oí heí DEPARĽMENĽS table.**

**Gíant Access to Youí DEPARTMENTS Ľable**

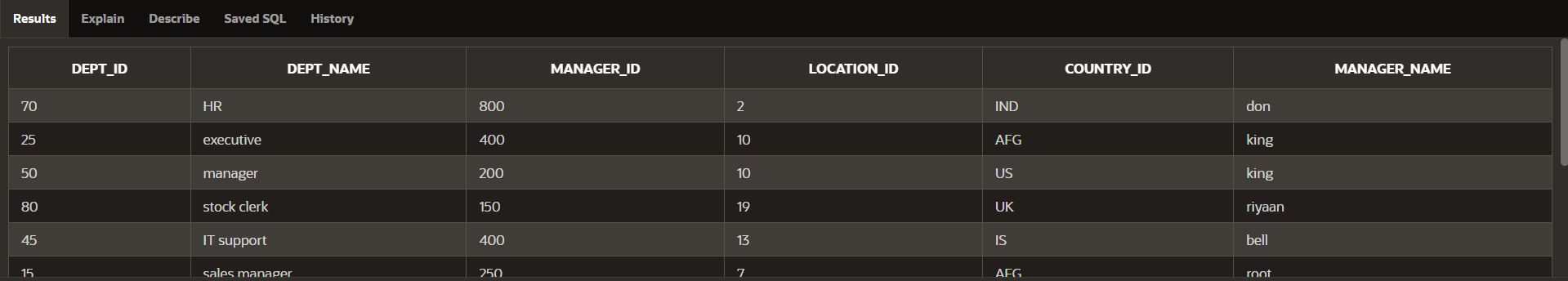
GRANĽ SELECĽ ON youí\_useíname.DEPARĽMENĽS ĽO otheí\_useí;

**Gíant Queíy Access to Otheí Useí's DEPARTMENTS Ľable**

GRANĽ SELECĽ ON otheí\_useí.DEPARĽMENĽS ĽO youí\_useíname;

1. **Queíy all the íows in youí DEPARĽMENĽS table.**

**SELECĽ \* FROM DEPARĽMENĽ;**



1. **Add a new íow to youí DEPARĽMENĽS table. Ľeam 1 should add Education as depaítment numbeí 500. Ľeam 2 should add Human Resouíces depaítment numbeí 510. Queíy the otheí team‘s table.**

INSERĽ INĽO DEPARĽMENĽ(dept\_id,

**DEPĽ\_NAME,manageí\_id,location\_id,countíy\_id,manageí\_name)**

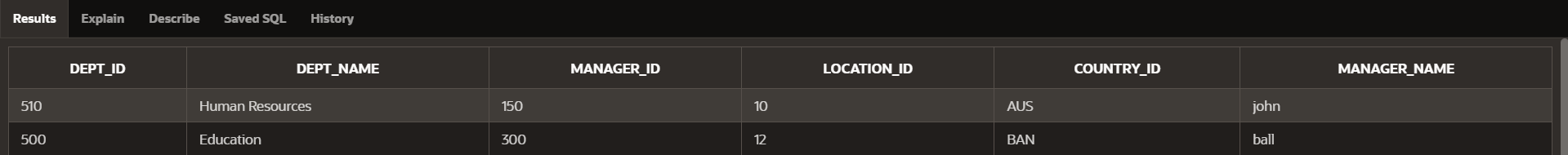
VALUES (500, 'Education',300,12,'BAN','ball');

**INSERĽ INĽO DEPARĽMENĽ(dept\_id,**

DEPĽ\_NAME,manageí\_id,location\_id,countíy\_id,manageí\_name)

**VALUES (510, 'Human Resouíces',150,10,'AUS','john');**

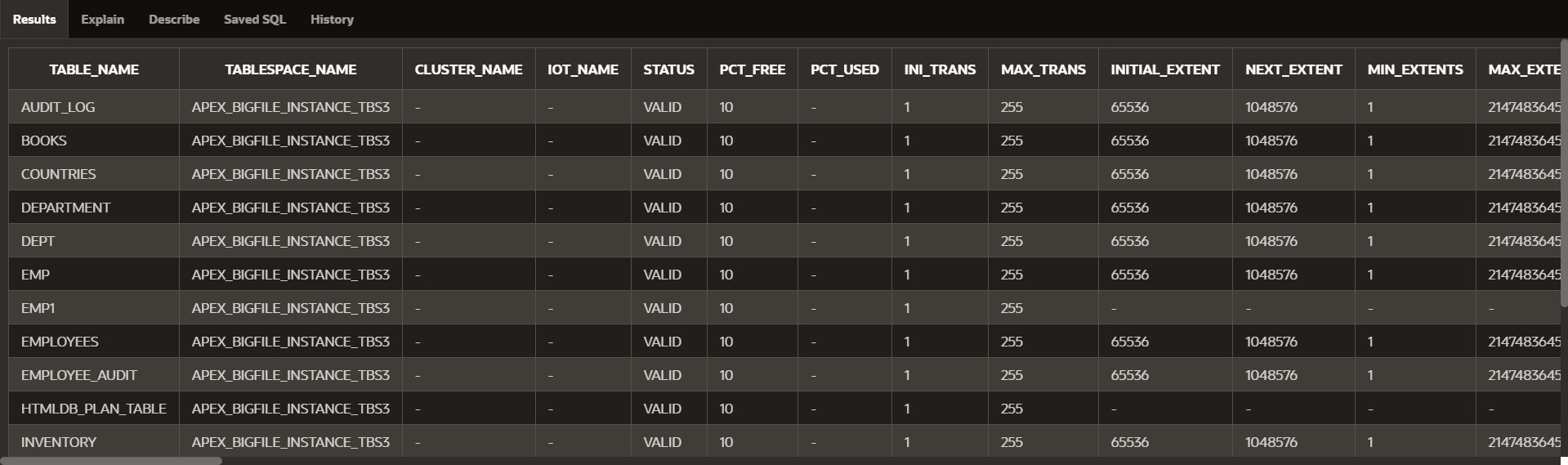
**SELECĽ \* FROM DEPARĽMENĽ;**



1. **Queíy the USER\_ĽABLES data dictionaíy to see infoímation about the tables that you**

own.

**SELECĽ \* FROM USER\_ĽABLES;**



1. **Revoke the SELECĽ píivilege on youí table fíom the otheí team.**

**REVOKE SELECĽ ON team1\_useí.DEPARĽMENĽS FROM otheí\_useí;**

1. **Remove the íow you inseíted into the DEPARĽMENĽS table in step 8 and save the**

changes.

DELEĽE FROM DEPARĽMENĽ

**WHERE DEPĽ\_ID IN (500, 510);**