**RAJALAKSHMI ENGINEERING COLLEGERAJALAKSHMINAGAR,THANDALAM– 602 105**

A white sign with blue text

Description automatically generated

CS23332DATABASEMANAGEMENT

SYSTEMSLAB

**Laboratory Record Notebook**

Name:

**DEVESH.D**

Year / Branch / Section:

**2nd year / B.Tech -AIML/ A**

UniversityRegisterNo:

**2116231501034**

CollegeRollNo:

**231501034**

Semester:

**3rd Semester**

AcademicYear:

**2023 - 2024**

**CS23332DATABASEMANAGEMENTSYSTEMS**

|  |  |
| --- | --- |
| **NAME** | **DEVESH D** |
| **ROLL NO.** | **2116231501034** |
| **DEPT** | **AIML** |
| **SEC** | **‘A’** |

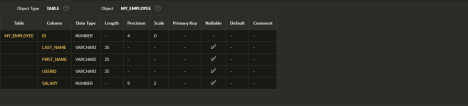
|  |  |  |
| --- | --- | --- |
| **Ex.No.: 1** | | **CREATIONOFBASETABLEAND**  **DML OPERATIONS** |
| **Date:** | 01/08/2024 |

1. CreateMY\_EMPLOYEEtablewiththefollowingstructure

CREATE TABLE MY\_EMPLOYEE(IDNumber(4)NOTNULL,

Last\_name Varchar(25),First\_name Varchar(25),Userid Varchar(25),SalaryNumber(9,2)

);



1. AddthefirstrowandsecondrowsdatatoMY\_EMPLOYEEtablefromthesampletable

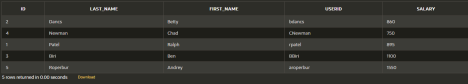
Insert intoMY\_EMPLOYEE(&ID,&LAST\_NAME,&FIRST\_NAME,&USERID,&SALARY

)

values(1,”Patel”,”Ralph”,”rpatel”,8952,”Dancs”,”Betty”,”bdancs”,860);

1. Display the table with values

Select\*fromMY\_EMPLOYEE;

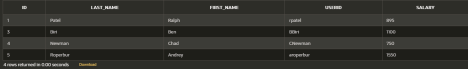


1. populate the next two rows of data from the sample data. Concatenatethefirstletterofthefirst\_NAMEwithfirstsevenlettersofthelast\_nametoproduceUserid

UpdateMY\_EMPLOYEES

SetUserid=substr(first\_name,1,1)||substr(last\_name,1,7)WhereIDin(3,4);

1. deleteBettydancsfrommy\_employeetable`1DeletefromMY\_EMPLOYEE

WhereFIRST\_NAME=‘Betty’andLAST\_NAME=‘Dancs’;

1. EmptythefourthrowoftheemptableDeletefromMY\_EMPLOYEE

WhereID=5;



1. MakethedataadditionspermanentCommit;
2. Changethelastnameofemployee3toDrexler

UpdateMY\_EMPLOYEESet LAST\_NAME = “Drexler”WhereID=3;



1. Changethesalaryto1000foralltheemployeeswithasalarylessthan900.

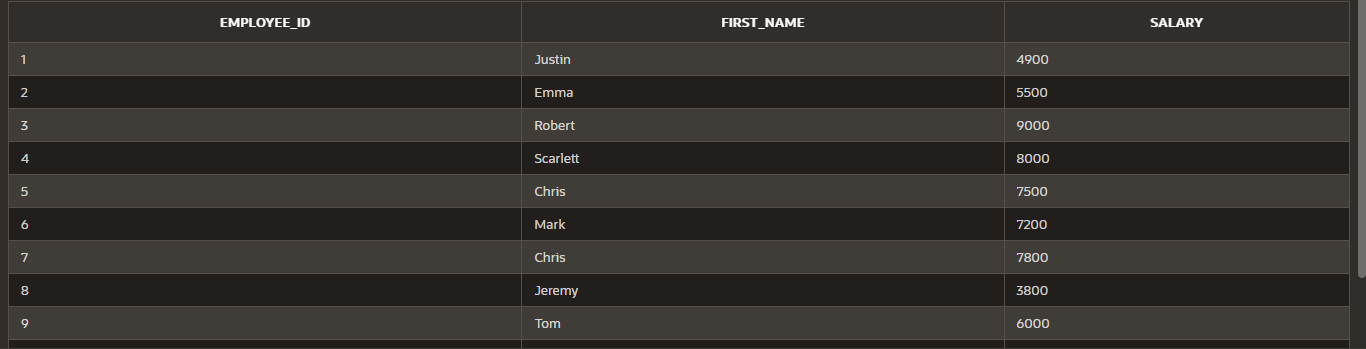
Update MY\_EMPLOYEESetsalary=1000

Wheresalary<900;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 2** | | **DATA MANIPULATIONS** |
| **Date:** | 08/08/2024 |

1. Findouttheemployeeid,names,salariesofalltheemployeesselectEmployee\_id,First\_Name,SalaryfromEMPLOYEES;



1. Listouttheemployeeswhoworksundermanager100

selectFirst\_Name||''||Last\_NameasnamefromEMPLOYEESwheremanager\_id

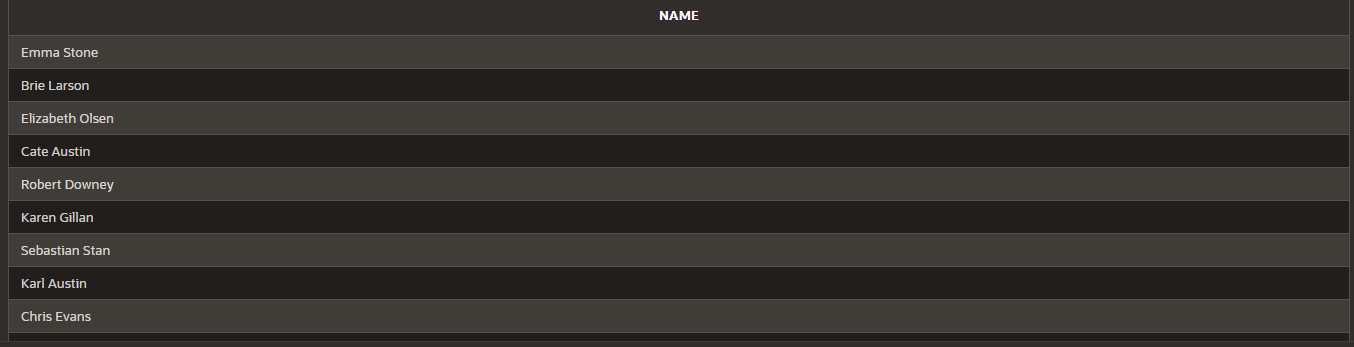
=100;



1. Findthenamesoftheemployeeswhohaveasalarygreaterthanorequalto4800

selectFirst\_Name ||''||Last\_NameasnamefromEMPLOYEES

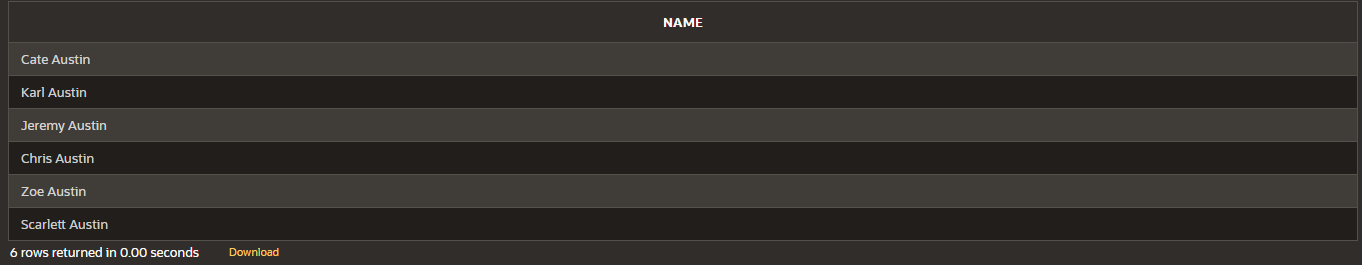
Wheresalary>=4800;



1. List out theemployeeswhose last nameis ̳AUSTIN

selectFirst\_Name||''||Last\_NameasnamefromEMPLOYEES

whereLast\_Name='Austin';



1. Findthenamesoftheemployeeswhoworksindepartments60,70and80

selectFirst\_Name||''||Last\_NameasnamefromEMPLOYEES

whereDepartment\_idin(60,70,80);



1. DisplaytheuniqueManager\_Id.

selectDISTINCT(manager\_id)fromEMPLOYEES;



1. InsertFiveRecordsandcalculateGrossPayandNetPay.

INSERTINTOEmp(EmpNo,EmpName,Job,Basic,DA,HRA,PF,GrossPay,NetPay)VALUES(

101,'JohnDoe','Manager',50000,15000,20000,6000,0,0,

102,'JaneSmith','Developer',40000,12000,16000,4800,0,0,

103,'AliceJohnson','Analyst',35000,10500,14000,4200,0,0,

104,'BobBrown','Designer',30000,9000,12000,3600,0,0,

105,'CharlieDavis','Tester',25000,7500,10000,3000,0,0

)

updateemp

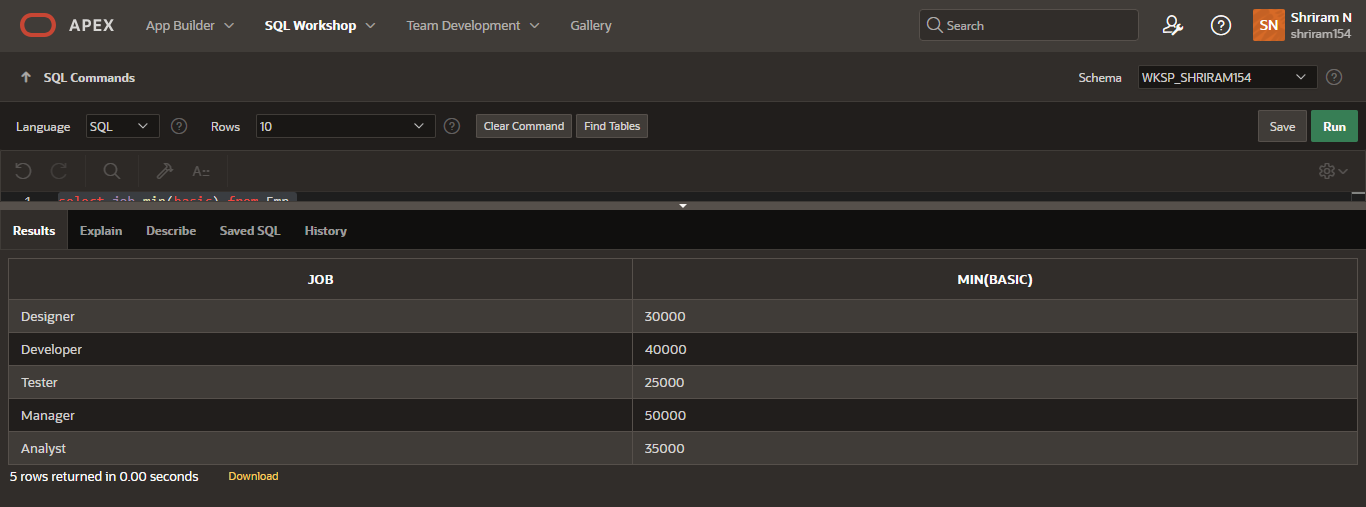
setGrossPay=Basic+DA+HRAwhereGrosspay=0;

updateemp

setNetPay=Grosspay-PFwhereNetpay=0;

1. DisplaytheemployeeswhoseBasicislowestineachdepartment.selectjob,min(basic) fromEmp

groupbyJob;



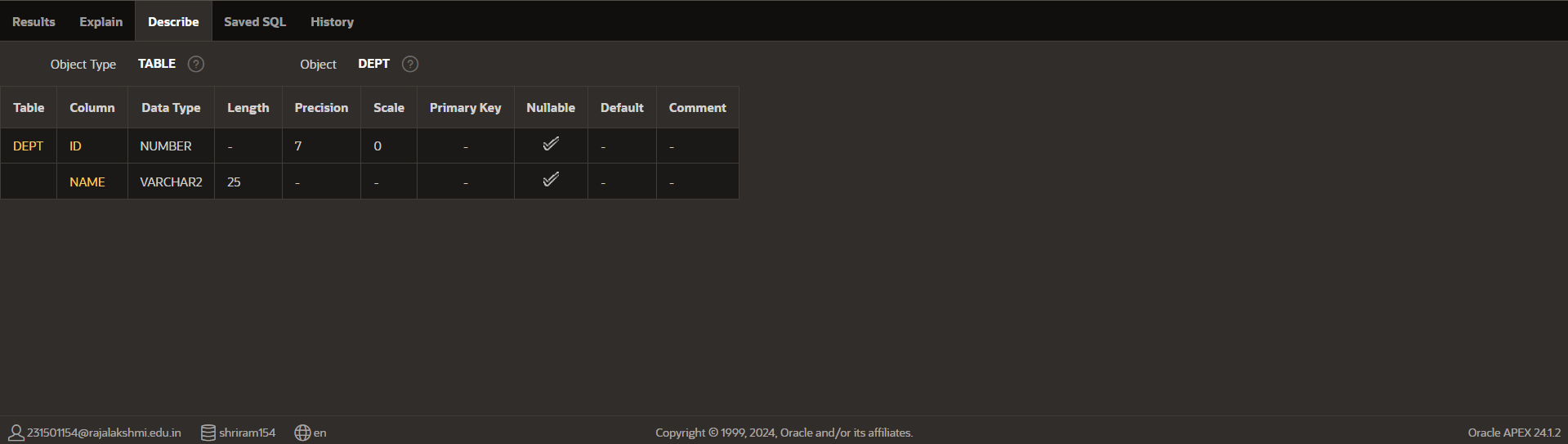
1. CreatetheDEPTtablebasedontheDEPARTMENTfollowingthetableinstancechartbelow.Confirmthatthetableiscreated.

CreatetableDEPT(

ID Number(7),Namevarchar(25)

);

DescDEPT;



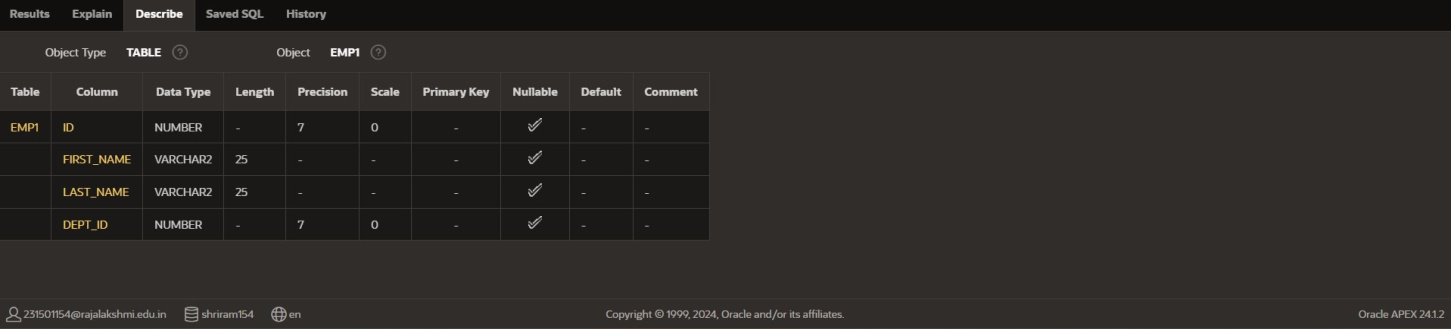
1. CreatetheEMP1tablebasedonthefollowinginstancechart.Confirmthatthetableiscreated.

create table EMP1(IDNumber(7),

First\_name varchar(25),Last\_name varchar(25),Dept\_idNumber(7)

);

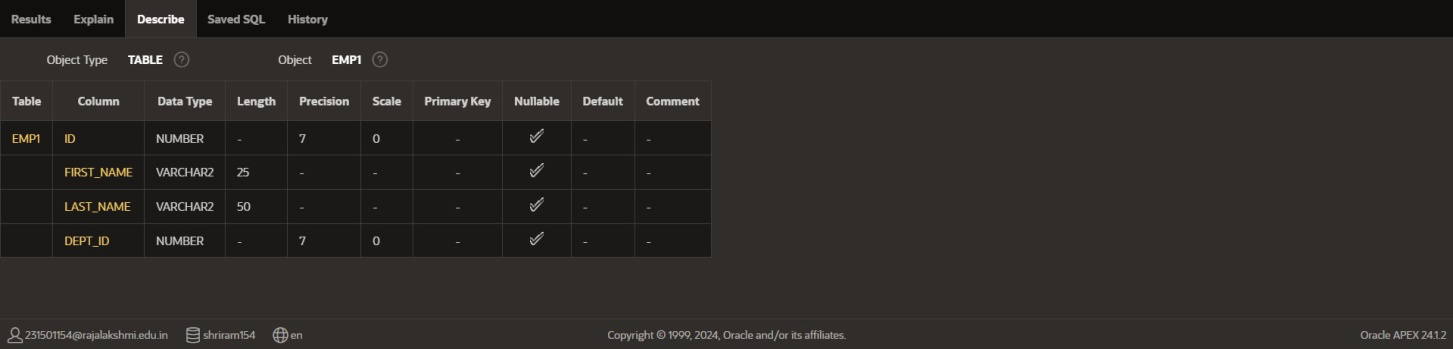
DescEMP1;



1. ModifytheEMP1tabletoallowforlongeremployeelastnames.Confirmthemodification.(Hint:Increasethesizeto50)

ALTERTABLEEMP1

modifyLast\_namevarchar(50);



1. Create the EMPLOYEES2 table based on the structure of EMPLOYEES table.IncludeOnlytheEmployee\_id,First\_name,Last\_name,SalaryandDept\_idcoloumns.NamethecolumnsId,First\_name,Last\_name,salaryandDept\_idrespectively.

create table EMPLOYEES2(ID Number(10),First\_name varchar(50),Last\_name varchar(50),Salary Number(10),Dept\_idNumber(10)

);

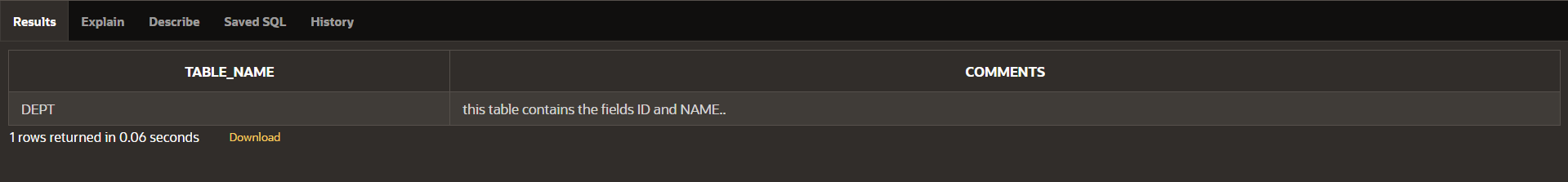
1. DroptheEMP1table.droptableEMP1;
2. RenametheEMPLOYEES2tableasEMP1.

ALTERTABLEEMPLOYEES2RENAMETOEMP1;

1. AddacommentonDEPTandEMP1tables.Confirmthemodificationbydescribingthetable.

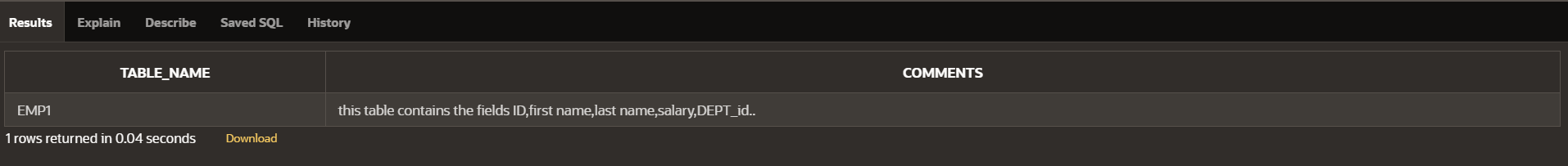
commentonTABLEDEPTIS'thistablecontainsthefieldsIDandNAME..';SELECTTABLE\_NAME,COMMENTS

FROMUSER\_TAB\_COMMENTS

WHERETABLE\_NAME='DEPT';

commentonTABLEEMP1IS'thistablecontainsthefieldsID,firstname,lastname,salary,DEPT\_id..';

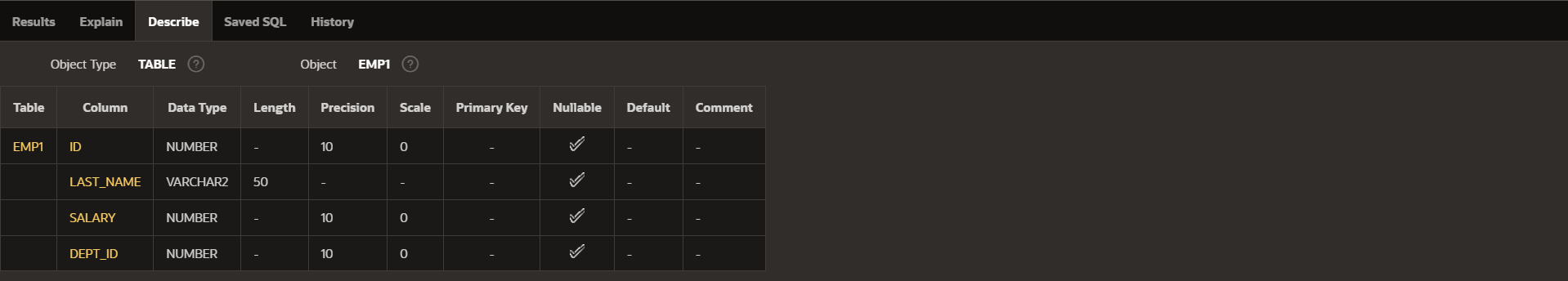
SELECT TABLE\_NAME, COMMENTSFROM USER\_TAB\_COMMENTSWHERETABLE\_NAME='EMP1';



1. DroptheFirst\_namecolumnfromtheEMPtableandconfirmit.

ALTERTABLEEMP1

dropcolumnFirst\_name;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 3** | | **WRITING BASIC SQL SELECT STATEMENTS** |
| **Date:** | 10/08/2024 |

FindtheSolutionforthefollowing:

TrueORFalse

1.Thefollowingstatementexecutessuccessfully.

IdentifytheErrors

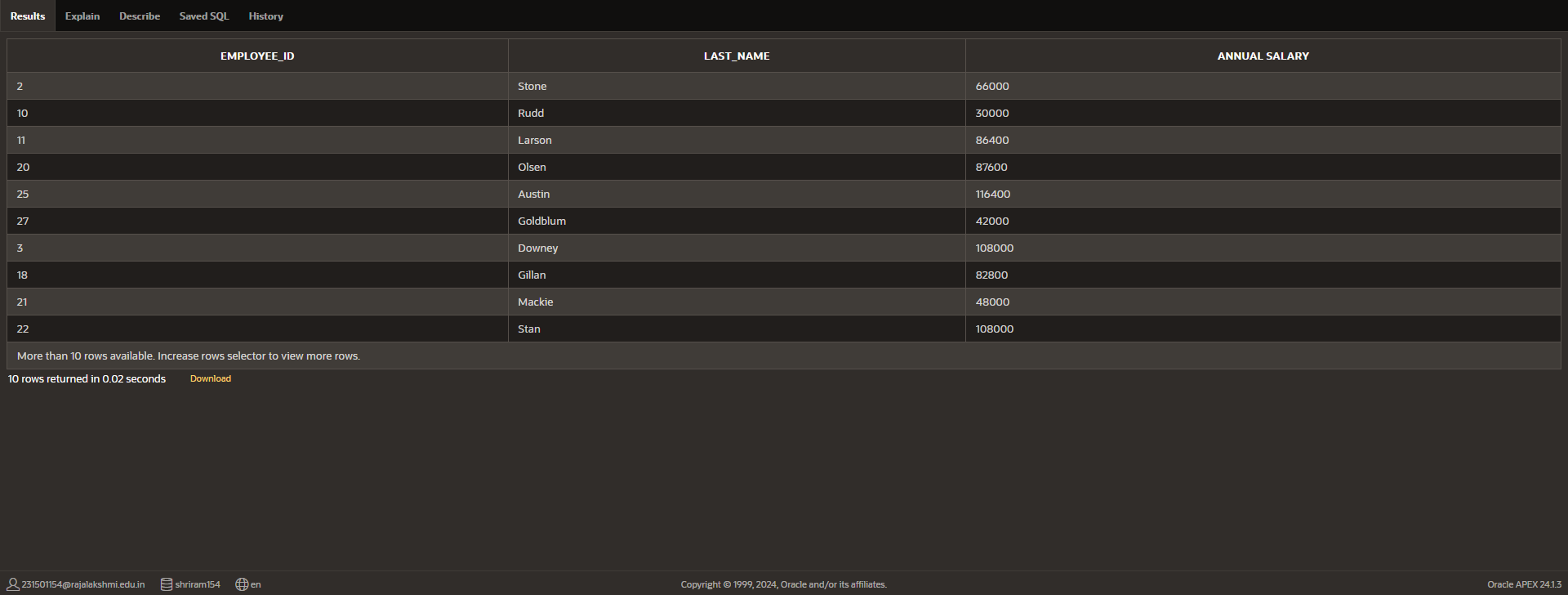
SELECT employee\_id, last\_namesal\*12ANNUAL SALARY

FROMemployees;

FALSE

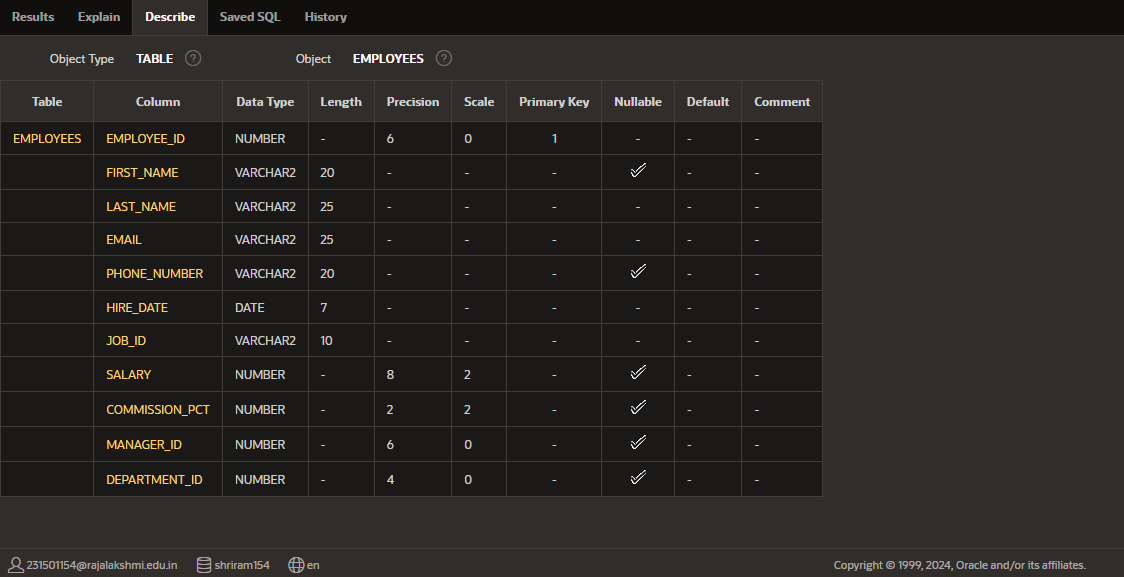
The columns in select statement should be separated by commas and the column alias shouldbegiven byusing akeyword“as”

SELECT employee\_id, last\_name, salary\*12 as "ANNUAL SALARY"FROMemployees;



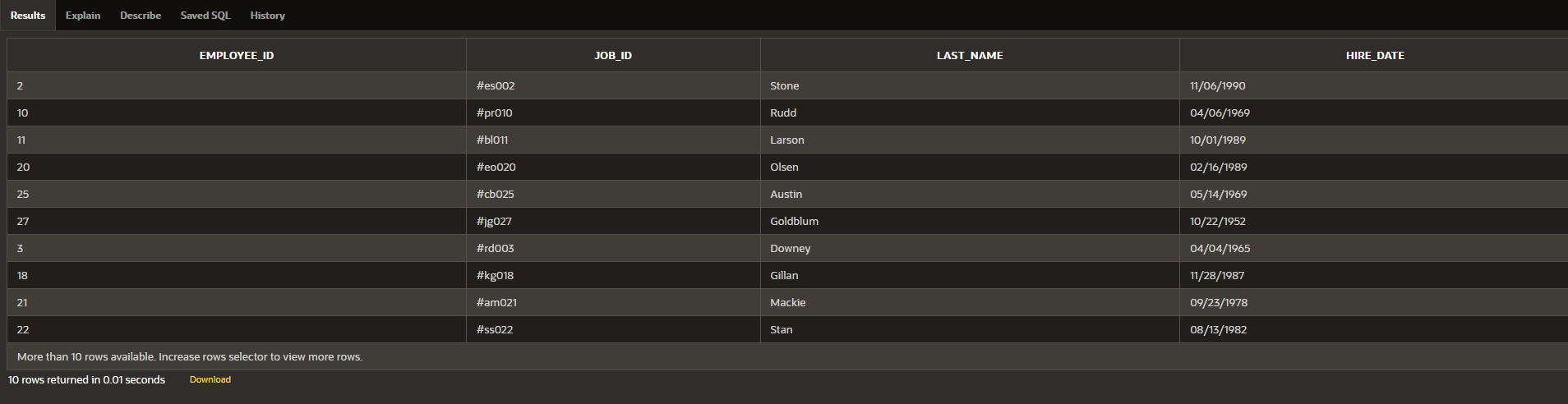
2)Showthestructureofdepartmentsthetable.Selectallthedatafromit.

Descemployees;

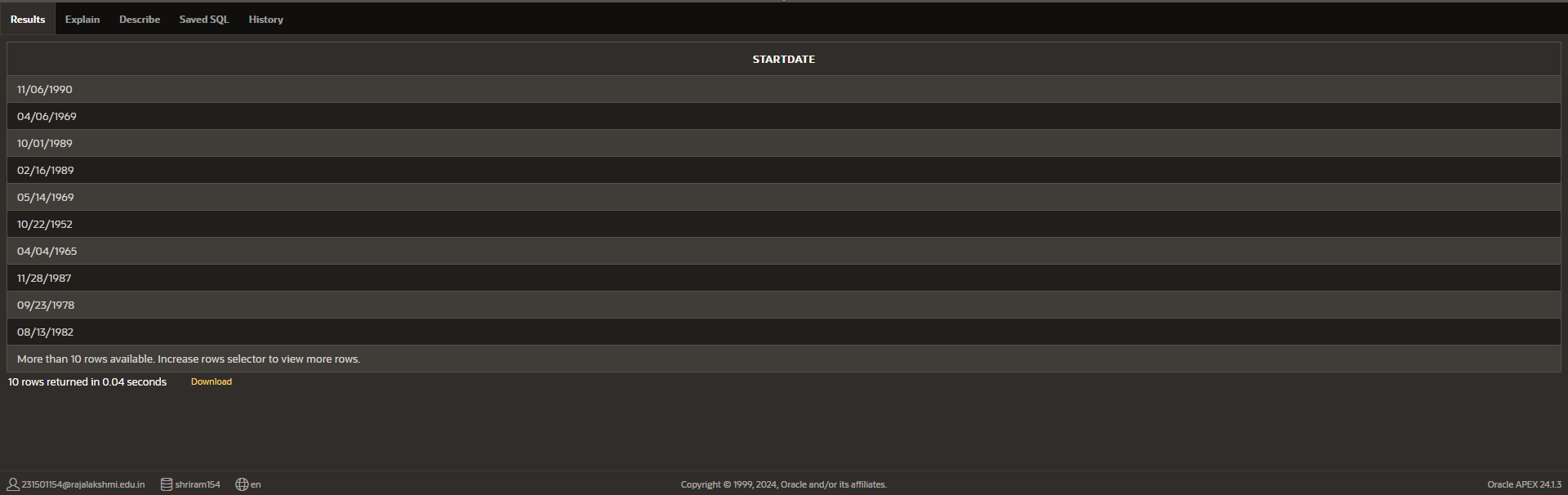


3. Create a query to display the last name, job code, hire date, and employee number foreachemployee, withemployee number appearingfirst.

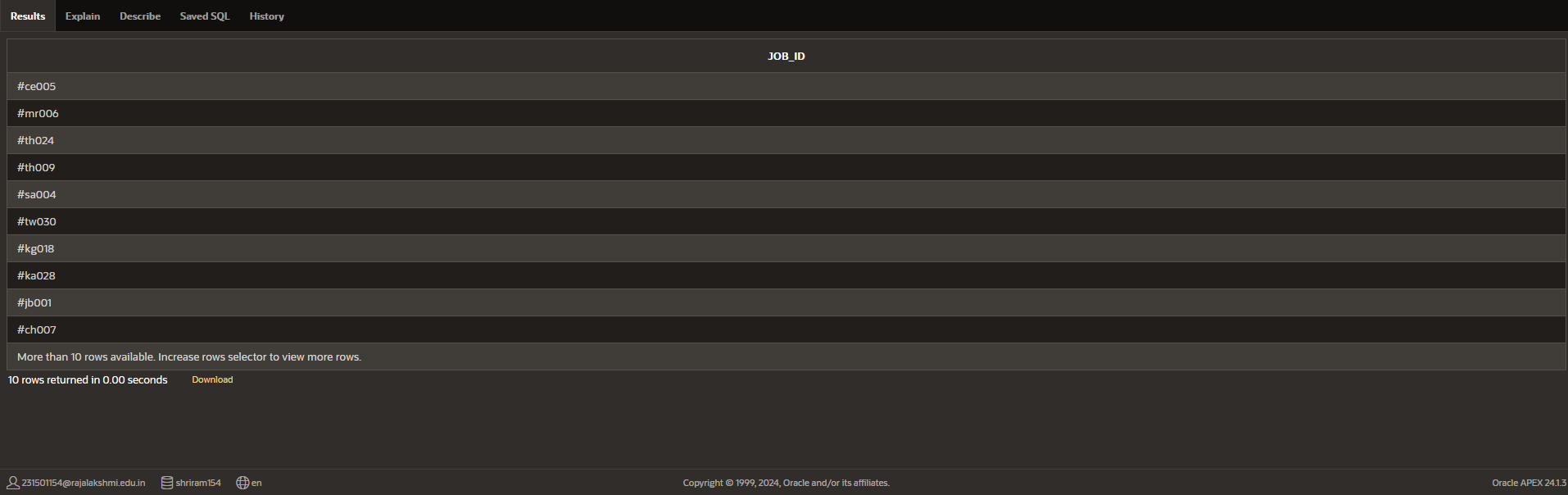
selectemployee\_id,job\_id,last\_name,hire\_datefromemployees;



1. Provide an alias STARTDATE for the hire date.selecthire\_dateas"STARTDATE"fromemployees;



1. Create a query to display unique job codes from the employee table.selectdistinct(job\_id) fromemployees;



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

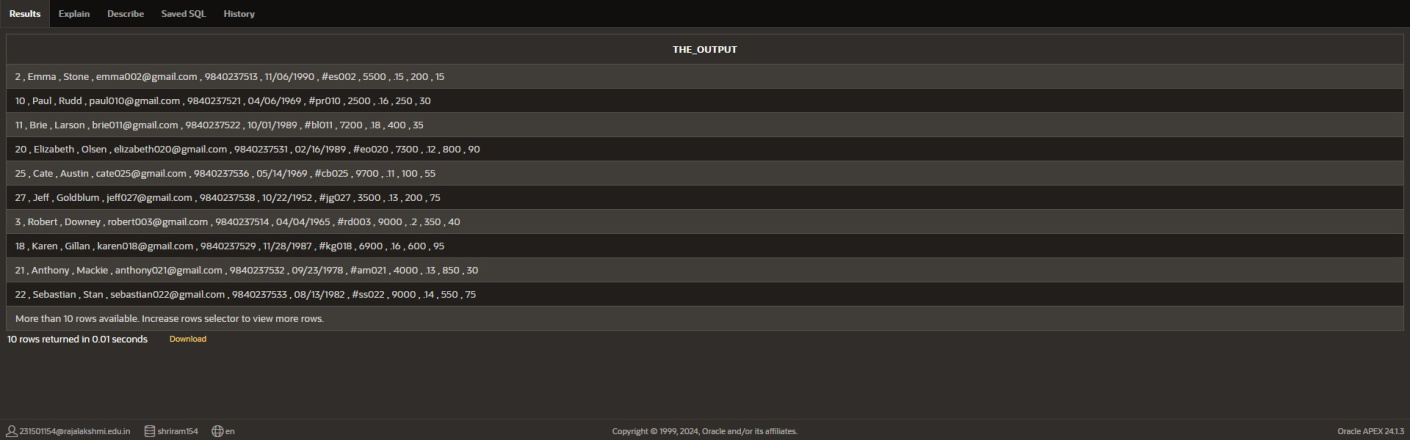
selectlast\_name||''||','||''||job\_idas"EMPLOYEEANDTITLE"fromemployees;



1. Create a query to display all the data from the employees table. Separate each column byacomma.Namethe columnTHE\_OUTPUT.

selectemployee\_id||','||first\_name||','||last\_name||','||email||','||phone\_number||',' || hire\_date || ' , ' || job\_id || ' , ' || salary || ' , ' || commission\_pct || ' , ' || manager\_id || ' , ' ||department\_idas"THE\_OUTPUT"

fromemployees;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 4** | | **WORKING WITH CONSTRAINTS** |
| **Date:** | 16/08/2024 |

* 1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column.Theconstraintshouldbenamedatcreation.Nametheconstraintmy\_emp\_id\_pk.

altertableEMP1

addconstraintmy\_emp\_id\_pkPRIMARYKEY(ID);

* 1. Create a PRIMAY KEY constraint to the DEPT table using the ID colum. The constraintshouldbenamedatcreation. Nametheconstraint my\_dept\_id\_pk.

altertableDEPT

addconstraintmy\_dept\_id\_pkPRIMARYKEY(ID);

* 1. Add a column DEPT\_ID to the EMP table. Add a foreign key reference on the EMP tablethat ensures that the employee is not assigned to nonexistent deparment. Name theconstraintmy\_emp\_dept\_id\_fk.

altertableemp

addDEPT\_IDNumbe(10);

altertableemp

addconstraintmy\_emp\_dept\_id\_fkFOREIGNKEY(DEPT\_ID)referencesdept(ID);

* 1. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision2, scale 2. Add a constraint to the commission column that ensures that a commissionvalueisgreaterthanzero.

altertableemp

addCOMMISSIONNumber(2,2);

altertableemp

addCONSTRAINTcommission\_gt\_zeroCHECK(COMMISSION>0);

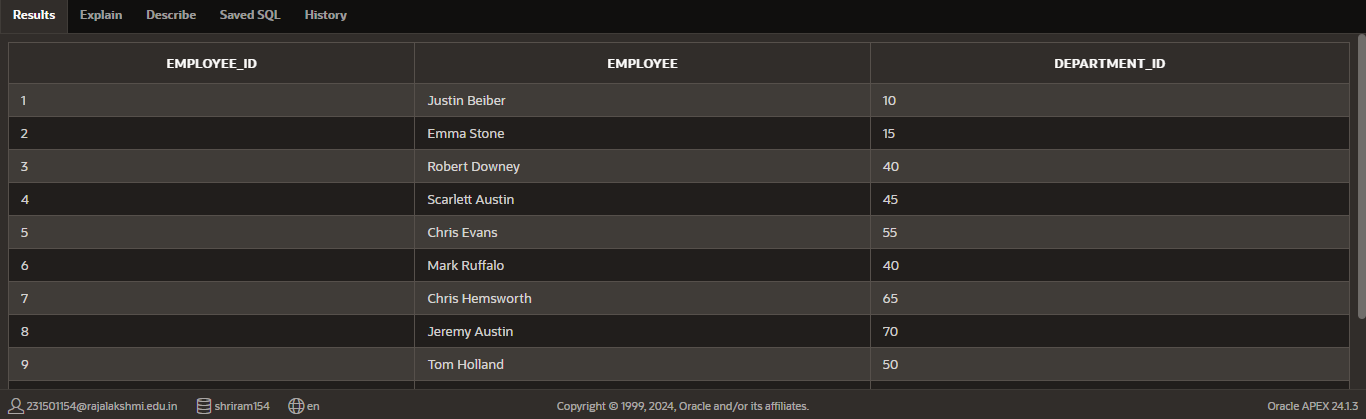
|  |  |  |
| --- | --- | --- |
| **Ex.No.: 5** | | **CREATINGVIEWS** |
| **Date:** | 23/08/2024 |

1. Create a view called EMPLOYEE\_VU based on the employee numbers, employeenames and department numbers from the EMPLOYEES table. Change the heading fortheemployee nametoEMPLOYEE.

createviewEMPLOYEE\_VUas

select employee\_id , first\_name || ' ' || last\_name as "EMPLOYEE", department\_idfromemployees;

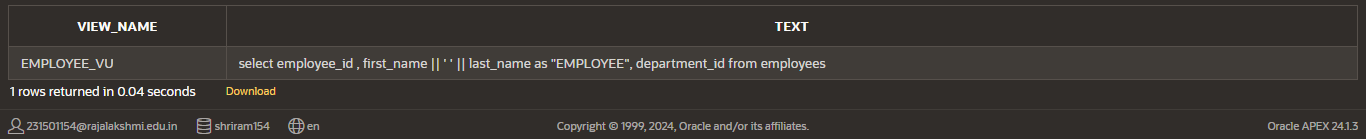
1. Display the contents of the EMPLOYEES\_VU view.select\*fromEMPLOYEE\_VU;



1. Select the view name and text from the USER\_VIEWS data dictionary views.selectVIEW\_NAME, TEXT

fromUSER\_VIEWS

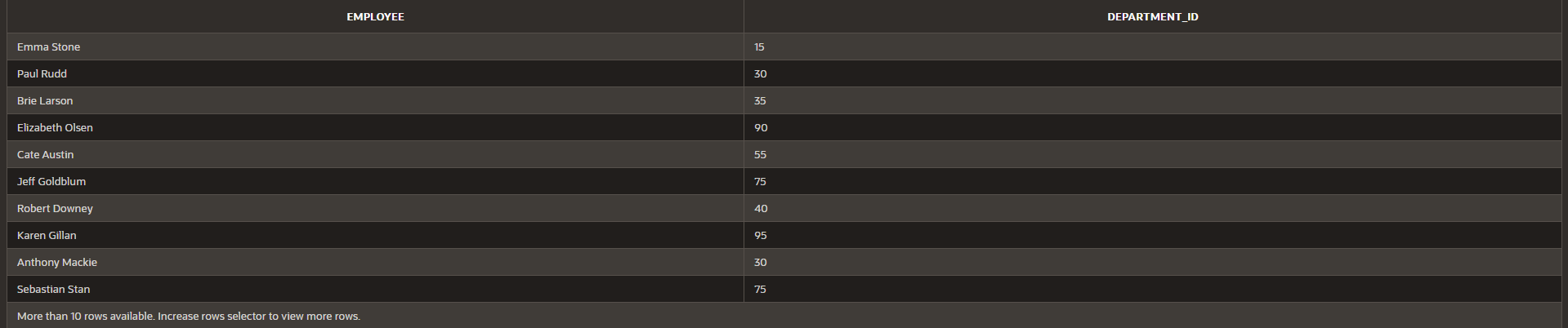
whereVIEW\_NAME='EMPLOYEE\_VU';



1. Using your EMPLOYEES\_VU view, enter a query to display all employees names andDepartment.

SELECTemployee,department\_id

FROMEMPLOYEE\_VU;



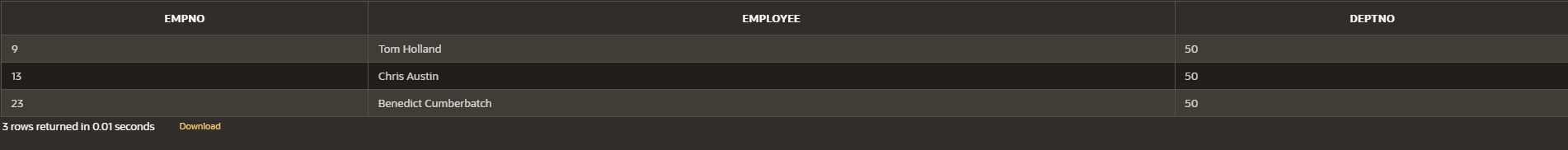
1. Create a view named DEPT50 that contains the employee number, employee lastnames and department numbers for all employees in department 50.Label the viewcolumns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to bereassignedtoanother department throughthe view.

CREATEVIEWDEPT50AS

SELECT employee\_id AS EMPNO,employee AS EMPLOYEE,department\_idASDEPTNO

FROMEMPLOYEE\_VU

WHERE department\_id = 50WITHREAD ONLY;



1. DisplaythestructureandcontentsoftheDEPT50view.

Descdept50;



1. AttempttoreassignMatostodepartment80.

UPDATEEMPLOYEES

SET department\_id = 80WHEREfirst\_name='Matos';

1. Create a view called SALARY\_VU based on the employee last names, departmentnames, salaries, and salary grades for all employees. Use the Employees,DEPARTMENTS and JOB\_GRADE tables. Label the column Employee, Department,salary,and Graderespectively.

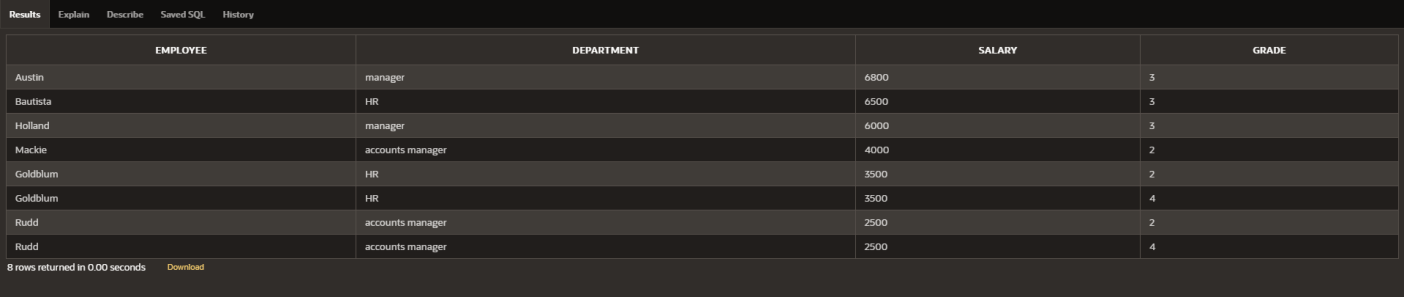
CREATEVIEWSALARY\_VU AS

SELECT e.last\_name AS Employee,d.dept\_name AS Department,e.salaryASSalary,j.grade\_levelAS Grade

FROM EMPLOYEES eJOINDEPARTMENTd

ON e.department\_id = d.dept\_idJOINJOB\_GRADE j

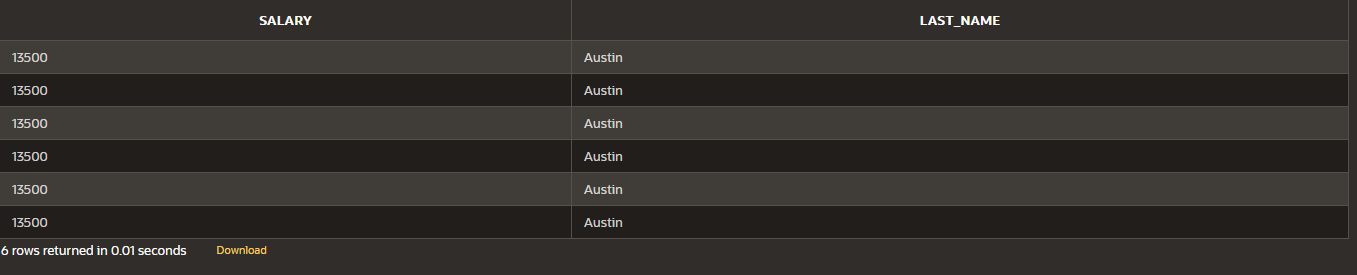
ONe.salaryBETWEENj.lowest\_salANDj.highest\_sal;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 6** | | **RESTRICTINGANDSORTINGDATA** |
| **Date:** | 29/08/2024 |

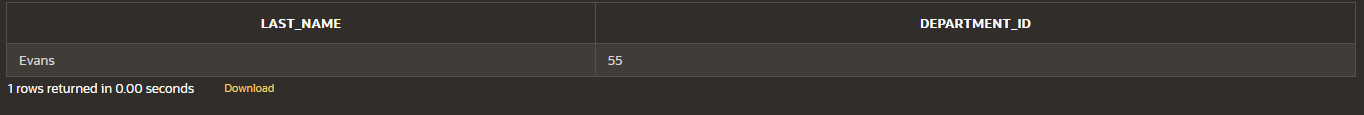
1. Create a query to display the last name and salary of employees earning more than12000.

selectsalary,last\_namefromemployeeswheresalary>12000;



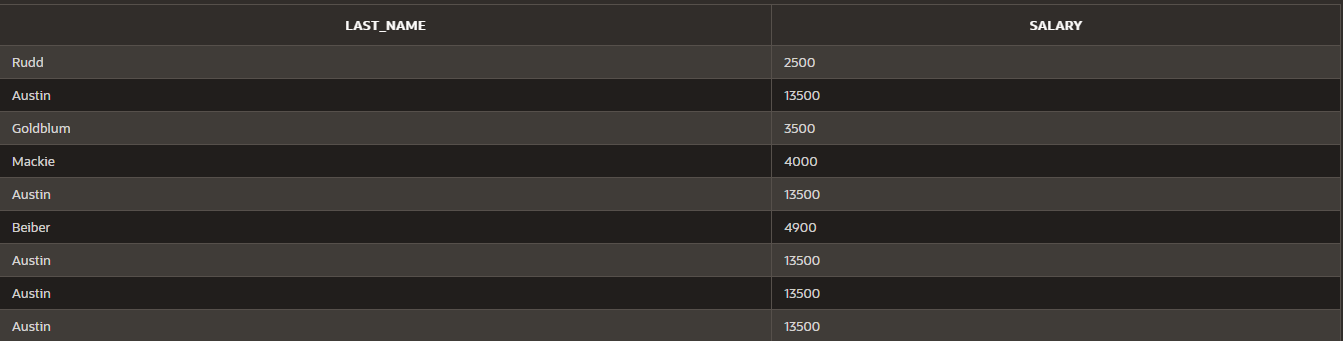
1. Create a query to display the employee last name and department number for employeenumber176.

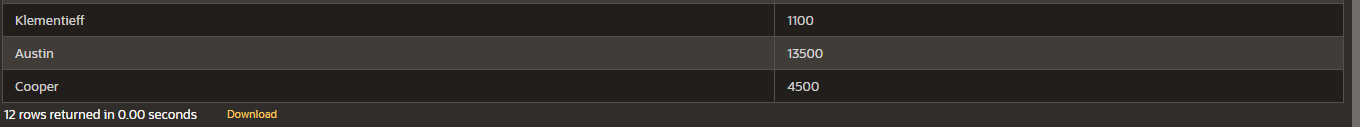
select last\_name , department\_id from employeeswhereemployee\_id =176;



1. Create a query to display the last name and salary of employees whose salary is not intherange of5000 and 12000.

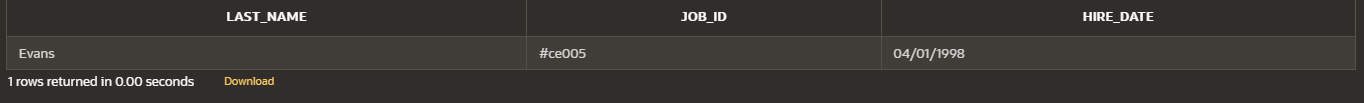
select last\_name , salary from employeeswheresalarynotbetween5000and12000;





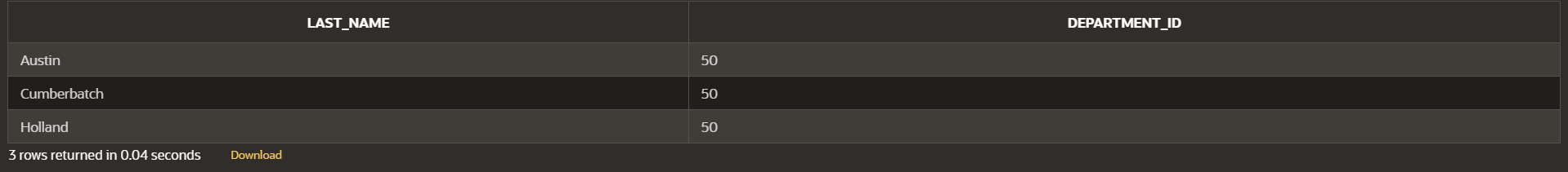
1. Display the employee last name, job ID, and start date of employees hired betweenFebruary 20,1998 and May 1,1998.order the query in ascending order by startdate.(hints:between)

select last\_name, job\_id, hire\_date from employeeswherehire\_datebetween'02-20-1998'and'05-01-1998';



1. Display the last name and department number of all employees in departments 20 and50inalphabetical orderbyname.

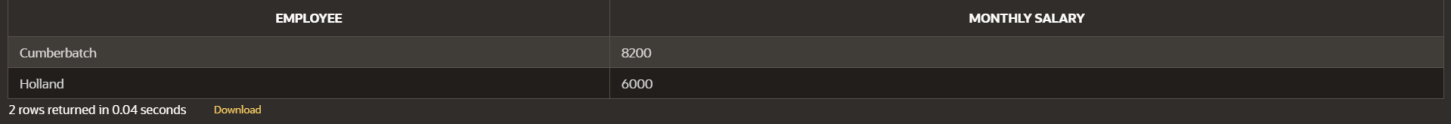
select last\_name, department\_id from employeeswhere department\_id = 20 or department\_id = 50orderbylast\_name;



1. Display the last name and salary of all employees who earn between 5000 and 12000and are in departments 20 and 50 in alphabetical order by name. Label the columnsEMPLOYEE,MONTHLY SALARY respectively.

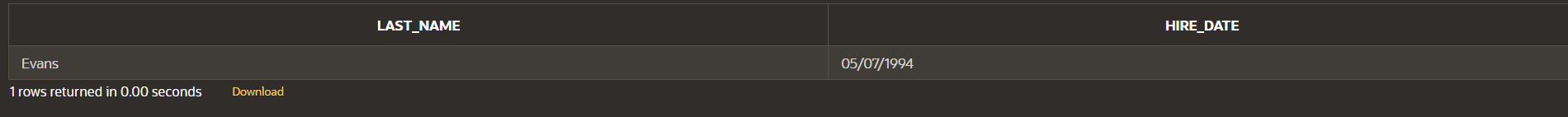
select last\_name as "EMPLOYEE" , salary as "MONTHLY SALARY" from employeeswheredepartment\_id in(20,50)and salarybetween 5000and 12000

orderbylast\_name;



1. Display the last name and hire date of every employee who was hired in 1994.selectlast\_name,hire\_date fromemployees

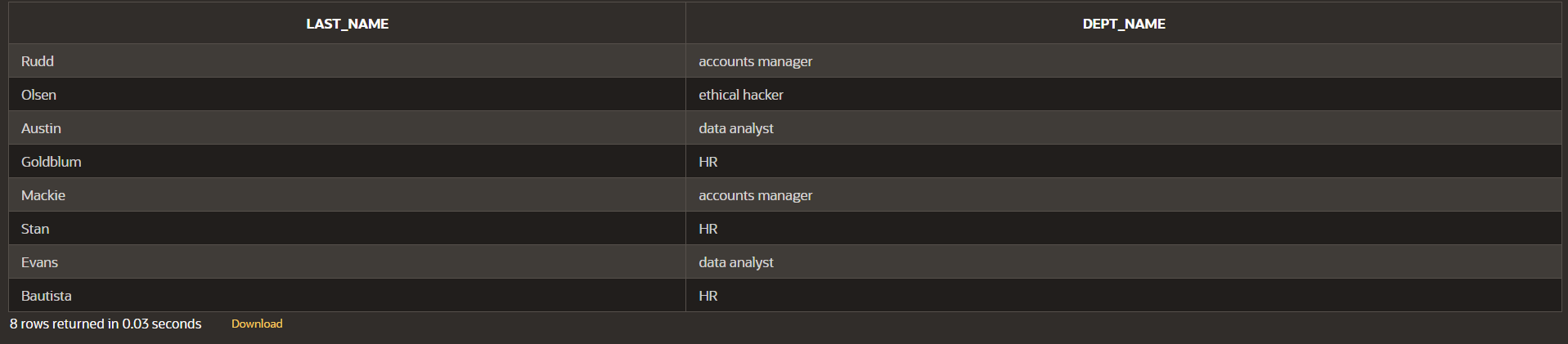
wherehire\_datelike'%1994%';



1. Displaythelastnameandjobtitleofallemployeeswhodonothaveamanager

select e.last\_name, d.dept\_name from employees ejoindepartment d

on e.department\_id = d.dept\_idwherenot(dept\_name='manager');



1. Display the last name, salary, and commission for all employees who earn commissions.Sortdataindescendingorderofsalaryandcommissions.(hints:isnotnul,orderby)

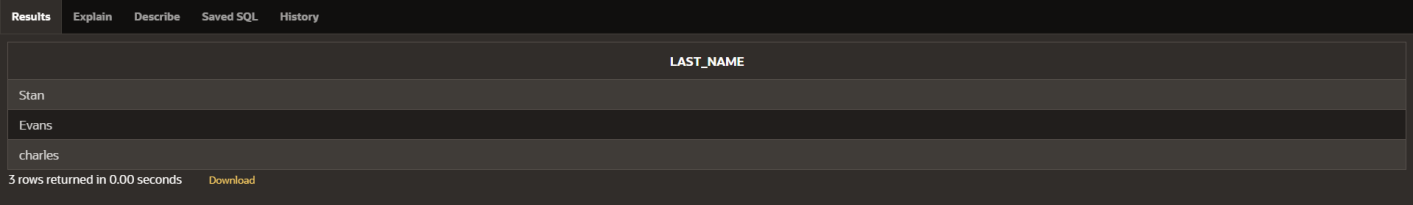
select last\_name,salary,commission\_pct from employeeswherecommission\_pct isnot null

orderbysalary,commission\_pctdesc;



1. Displaythelastnameofall employeeswherethethirdletterofthenameisa.

selectlast\_namefromemployeeswherelast\_name like'a%';



1. Displaythelastnameofallemployeeswhohaveanaandaneintheirlastname.

SELECTlast\_nameFROMemployees

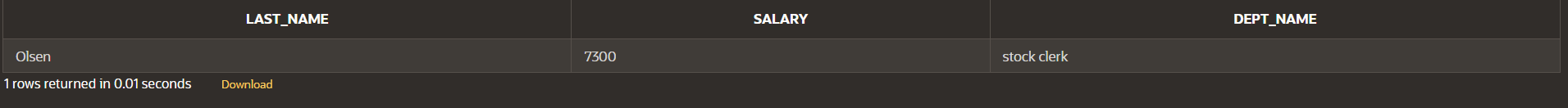
WHERElast\_nameLIKE'%a%'ANDlast\_nameLIKE'%e%';



1. Display the last name and job and salary for all employees whose job is salesrepresentativeorstockclerkandwhosesalaryisnotequalto2500,3500or7000/.

SELECT e.last\_name,e.salary,d.dept\_name FROM employees ejoindepartment done.department\_id=d.dept\_id

WHERE (dept\_name in ('stock clerk','sales representative')) and (salary notin(2500,3500,7000));

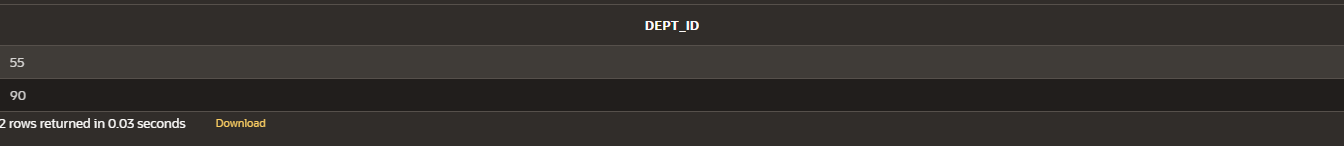


|  |  |  |
| --- | --- | --- |
| **Ex.No.: 7** | | **USING SET OPERATORS** |
| **Date:** | 30/08/2024 |

1. TheHRdepartmentneedsalistofdepartmentIDsfordepartmentsthatdonotcontainthejob IDST\_CLERK. Usesetoperatorstocreatethisreport.

select dept\_id from departmentminus

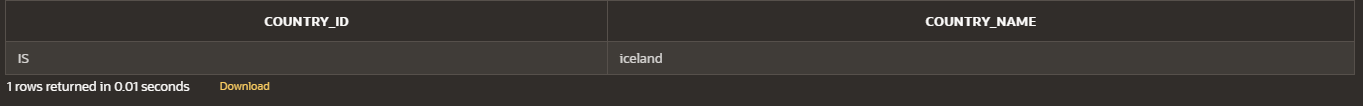
select department\_id from employeeswherejob\_id ='ST\_CLERK';



1. The HR department needs a list of countries that have no departments located inthem.Display the country ID and the name of the countries. Use set operators to createthisreport.

SELECT c.country\_id, c.country\_nameFROMcountries c

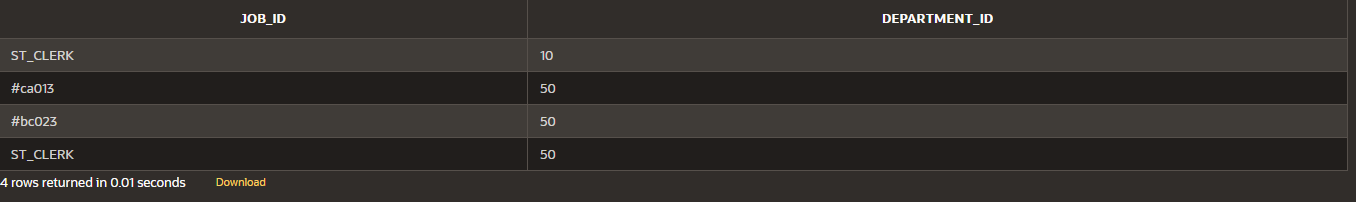
LEFTJOINdepartmentdONc.country\_id=d.country\_idWHEREd.country\_id ISNULL;



1. Producealistofjobsfordepartments10,50,and20,inthatorder.DisplayjobIDanddepartmentIDusing setoperators.

SELECT job\_id, department\_idFROMemployees

WHERE department\_id IN (10, 50, 20)ORDERBY department\_id;



1. Create a report that lists the employee IDs and job IDs of those employees whocurrently have a job title that is the same as their job title when they were initiallyhired by the company (that is, they changed jobs but have now gone back to doingtheiroriginal job).

SELECT employee\_id, job\_idFROM employeesINTERSECT

SELECT employee\_id, job\_idFROMjob\_history;



1. TheHRdepartmentneedsareportwiththefollowingspecifications:
   * Last name and department ID of all the employees from the EMPLOYEES table,regardlessofwhetherornot theybelong toadepartment.
   * Department ID and department name of all the departments from the DEPARTMENTStable, regardless of whether or not they have employees working in them Write acompoundquerytoaccomplish this.

SELECT last\_name, department\_id FROM employeesUNION

SELECTdept\_name,dept\_idFROMdepartment;



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 8** | | **WORKING WITH MULTIPLE TABLES** |
| **Date:** | 05/09/2024 |

1. Writeaquerytodisplaythelastname,departmentnumber,anddepartmentnameforallEmployees.

select e.last\_name , e.department\_id , d.dept\_namefromemployeese

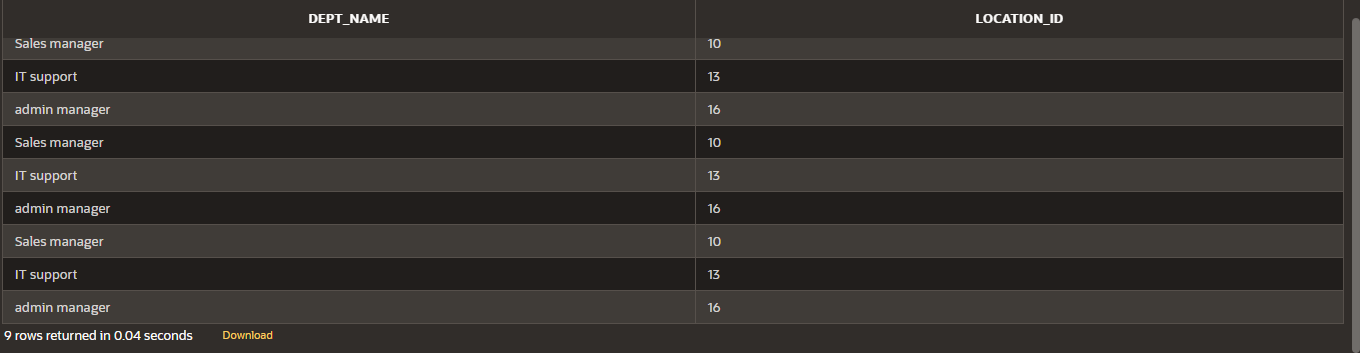
joindepartmentdone.department\_id=d.dept\_id;



1. Create a unique listing of all jobs that are in department 80. Include the location of thedepartmentintheoutput.

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

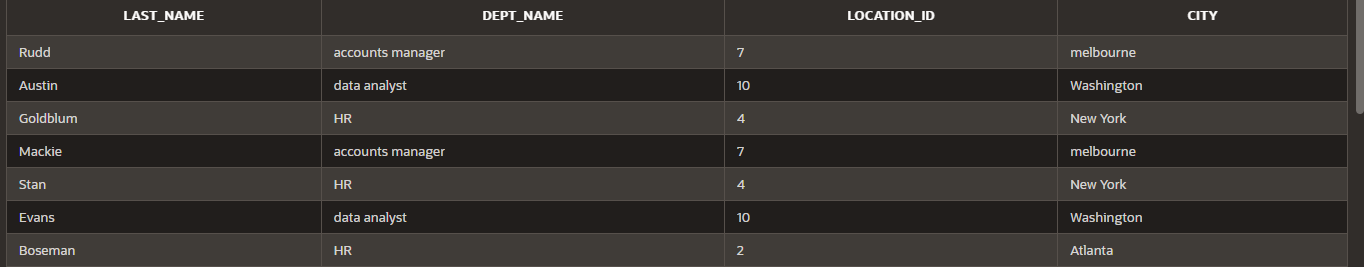
join employees e on d.dept\_id = e.department\_idwheredepartment\_id =80;



1. Write a query to display the employee last name, department name, location ID, and cityofall employees who earnacommission

select e.last\_name,d.dept\_name,d.location\_id,l.cityfrom(departmentd

inner join employees e on d.dept\_id = e.department\_idinner join location l on d.location\_id = l.location\_id)wherecommission\_pct isnot null;

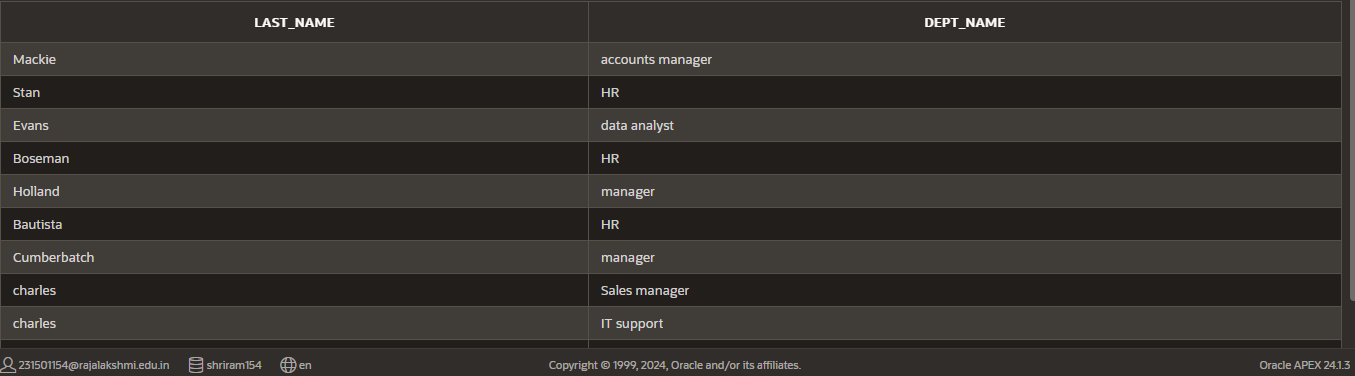




1. Display the employee last name and department name for all employees who have ana(lowercase)intheir last names.

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

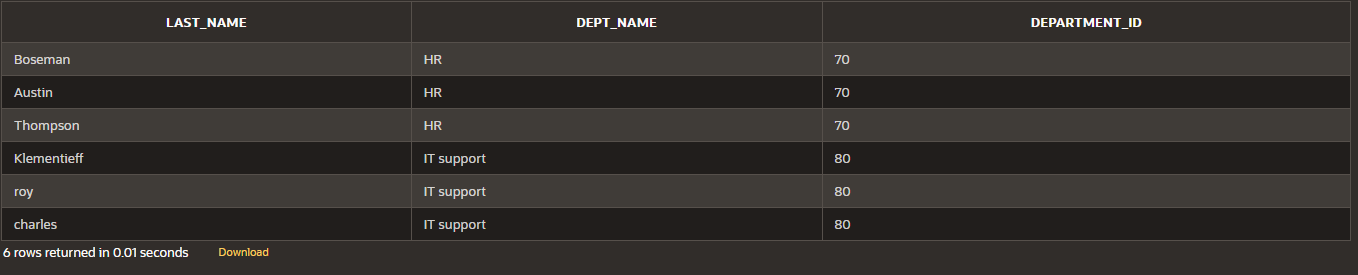
inner join employees e on d.dept\_id = e.department\_idwherelast\_name like'%a%';



1. Write a query to display the last name, job, department number, and department nameforall employees who workinToronto.

select e.last\_name,d.dept\_name,e.department\_idfrom(departmentd

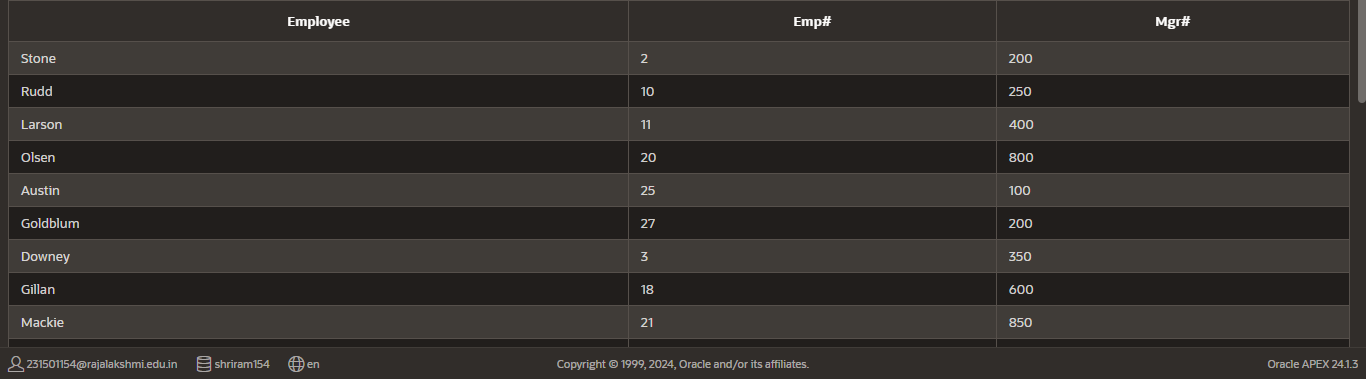
inner join employees e on d.dept\_id = e.department\_idinner join location l on l.location\_id = d.location\_id)wherecity='Toronto';



1. Display the employee last name and employee number along with their manager‘s lastnameandmanagernumber.LabelthecolumnsEmployee,Emp#,Manager,andMgr#,

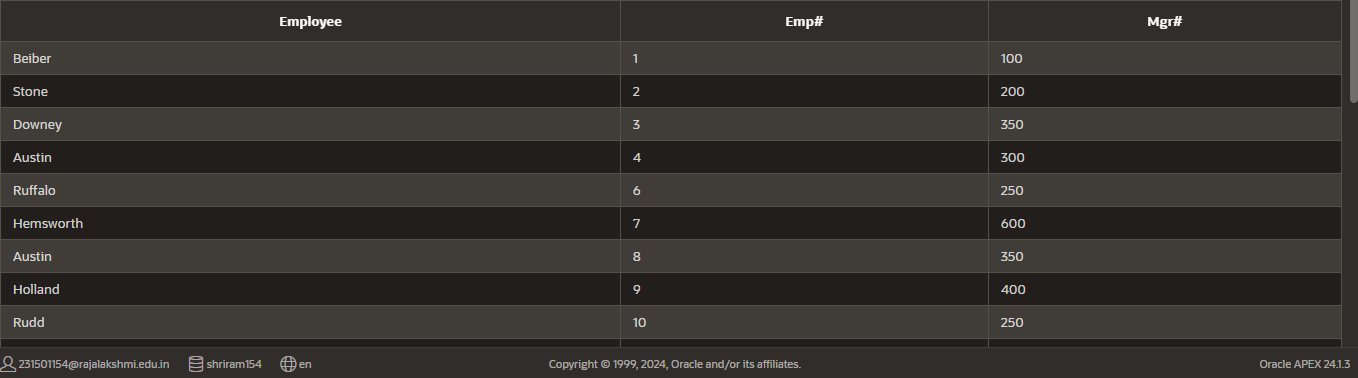
Respectively

select last\_name as "Employee",employee\_id as "Emp#",manager\_id as "Mgr#" fromemployees;



1. Modify lab4\_6.sql to display all employees including King, who has no manager. Ordertheresultsbytheemployee number.

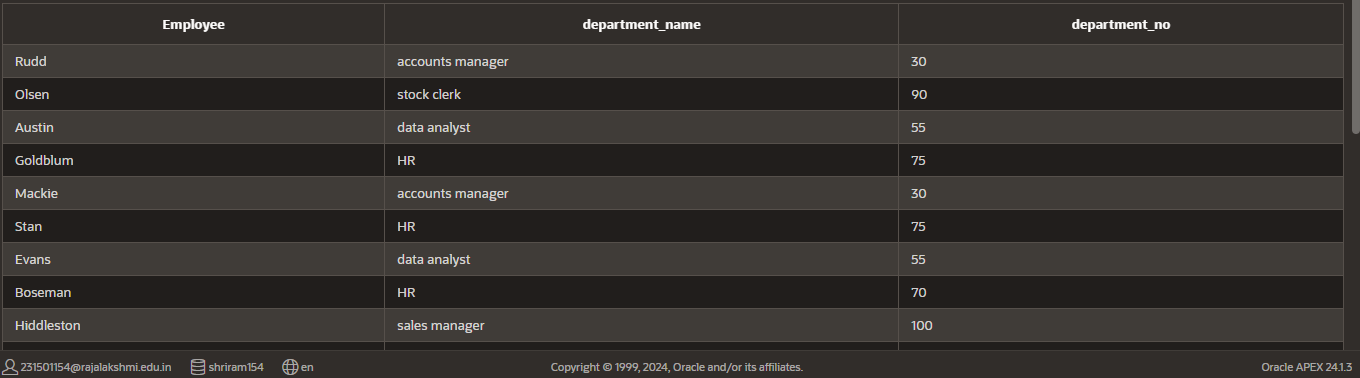
SELECT last\_name AS "Employee",employee\_id AS "Emp#",manager\_id AS "Mgr#"FROMemployeesORDERBYemployee\_id;



1. Create a query that displays employee last names, department numbers, and all theemployees who work in the same department as a given employee. Give each columnanappropriate label

select e.last\_name as "Employee",d.dept\_name as "department\_name",e.department\_idas"department\_no"fromemployeese

innerjoindepartmentdone.department\_id=d.dept\_id;



1. ShowthestructureoftheJOB\_GRADEStable.Createaquerythatdisplaysthename,job,

departmentname,salary,andgradeforallemployees

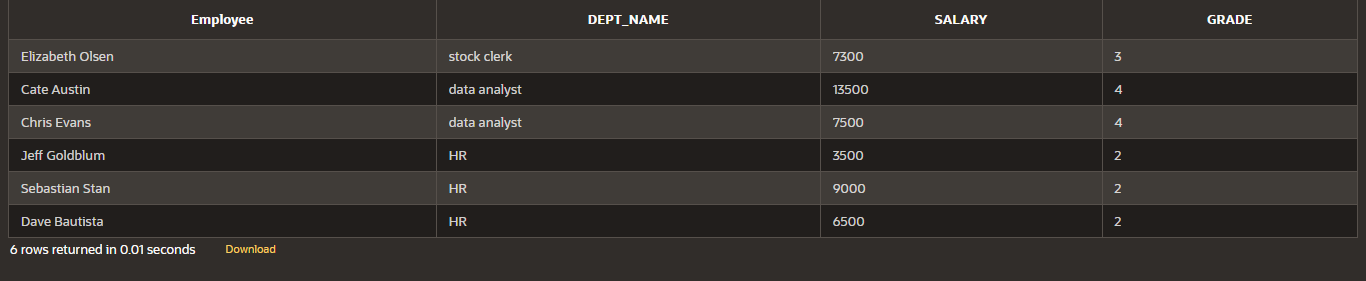
descjob\_grade;

SELECT e.first\_name || ' ' || last\_name AS "Employee",d.dept\_name,e.salary,g.grade\_level as"GRADE"

FROM(employeese

innerjoindepartmentdone.department\_id=d.dept\_id

innerjoinjob\_gradegone.department\_id=g.department\_id);



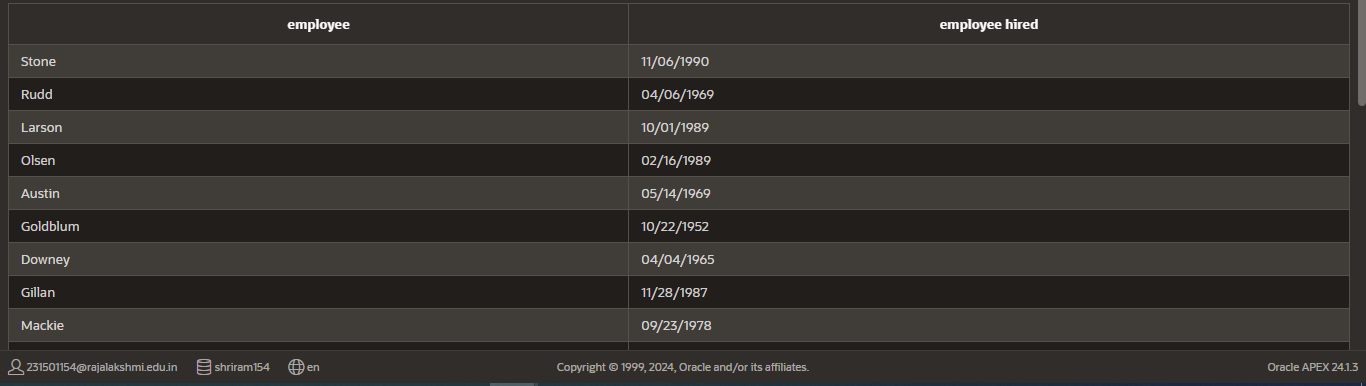
1. Create a query to display the name and hire date of any employee hired after employeeDavies.

SELECT last\_name,hire\_date FROM employeeswherehire\_date >'05-03-1986';



1. Display the names and hire dates for all employees who were hired before their managers,along with their manager‘s names and hire dates. Label the columns Employee, Emp Hired,Manager,and MgrHired, respectively.

SELECTlast\_nameas"employee",hire\_dateas"employeehired"FROMemployees;



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

* 1. The HR department needs a query that prompts the user for an employee last name.The query then displays the last name and hire date of any employee in the samedepartment as the employee whose name they supply (excluding that employee). Forexample, if the user enters Zlotkey, find all employees who work with Zlotkey (excludingZlotkey).

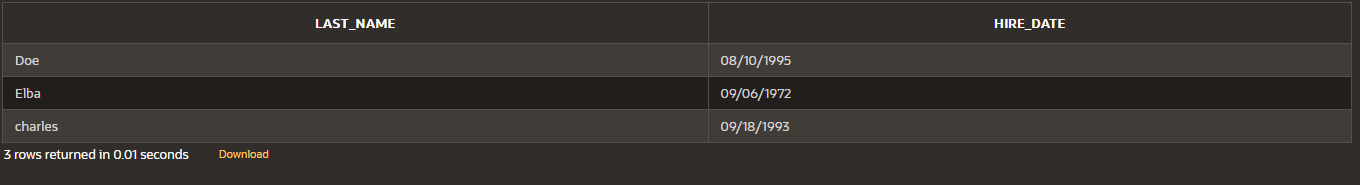
SELECT last\_name, hire\_dateFROMemployees

WHERE department\_id = ALL(SELECT department\_idFROMemployees

WHERElast\_name='Zlotkey'

)

ANDlast\_name!='Zlotkey';



* 1. Createareportthatdisplaystheemployeenumber,lastname,andsalaryofallemployees who earn more than the average salary. Sort the results in order ofascendingsalary.

SELECTEMPLOYEE\_ID,LAST\_NAME,SALARY

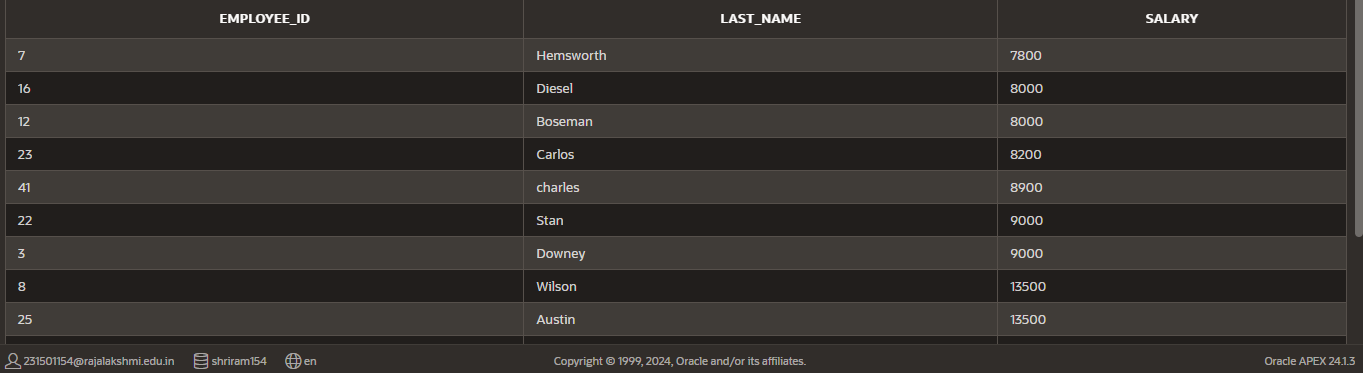
FROM employeesWHERESALARY>(

SELECTAVG(SALARY)

FROMemployees

)

ORDERBYSALARY ASC;



* 1. Write a query that displays the employee number and last name of all employees whoworkinadepartment withanyemployee whose lastnamecontainsau.

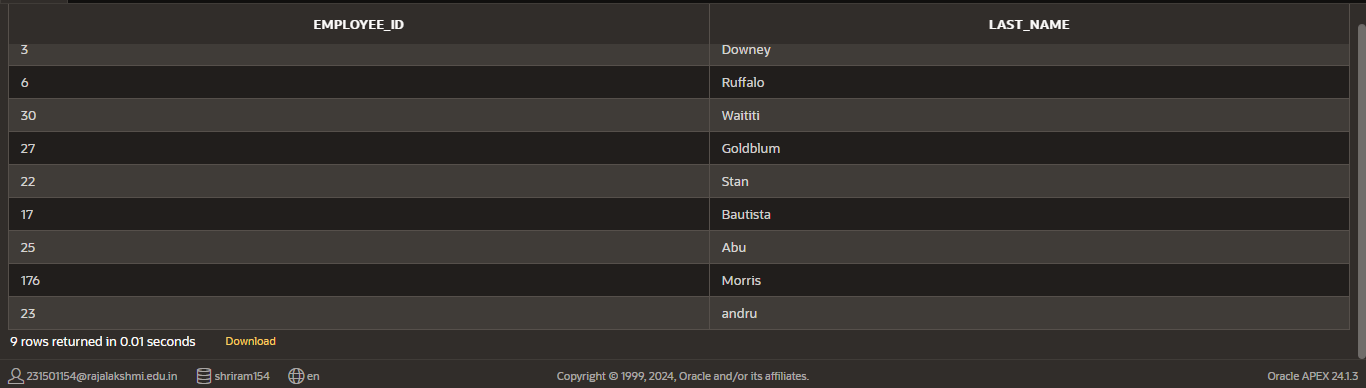
SELECTEMPLOYEE\_ID,LAST\_NAME

FROMemployees

WHERE DEPARTMENT\_ID IN (SELECTDEPARTMENT\_ID

FROMemployees

WHERELAST\_NAMELIKE'%a%'andLAST\_NAMELIKE'%u%');



* 1. The HR department needs a report that displays the last name, department number, andjobIDofall employeeswhose department location IDis1700.

SELECTe.last\_name,e.department\_id,e.job\_id

FROMemployeese

INNER JOIN department d ON e.department\_id = d.dept\_idWHEREe.department\_id IN(

SELECT dept\_idFROMdepartment

WHERElocation\_id=1700);

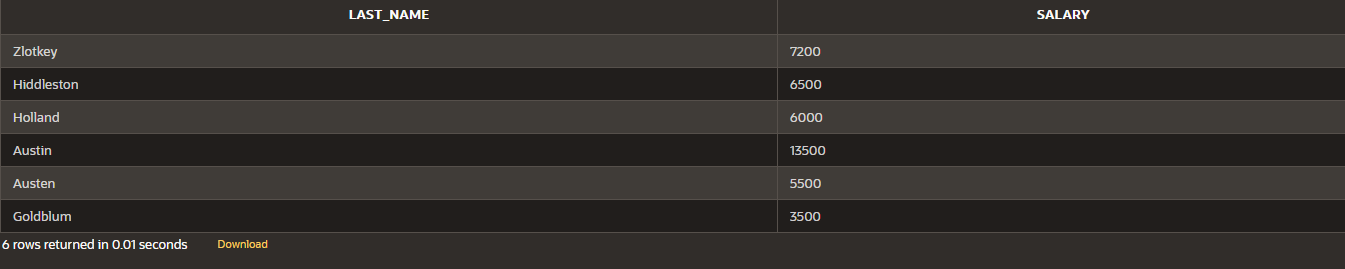


* 1. Create a report for HR that displays the last name and salary of every employee whoreportstoKing.

SELECT e.last\_name, e.salaryFROMemployeese

WHERE e.manager\_id IN (SELECT d.manager\_idFROMdepartment d

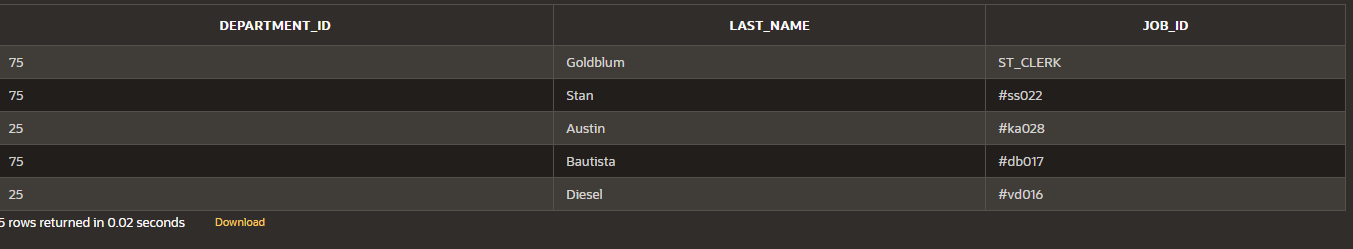
WHEREd.manager\_name='king');



* 1. CreateareportforHRthatdisplaysthedepartmentnumber,lastname,andjobIDforeveryemployee intheExecutive department.

SELECT e.department\_id, e.last\_name, e.job\_idFROMemployeese

JOIN department d on e.department\_id = d.dept\_idWHEREd.dept\_name ='executive';



* 1. Modify the query 3 to display the employee number, last name, and salary of allemployees who earn more than the average salary and who work in a department withanyemployee whose lastnamecontainsa u.

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

WHERE e.salary > (SELECT AVG(salary)FROMemployees

)

AND e.department\_id IN (SELECT x.department\_idFROMemployeesx

WHEREx.last\_nameLIKE'%a%'ANDx.last\_nameLIKE'%u%'

);



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 10** | | **AGGREGATINGDATAUSINGGROUPFUNCTIONS** |
| **Date:** | 12/09/2024 |

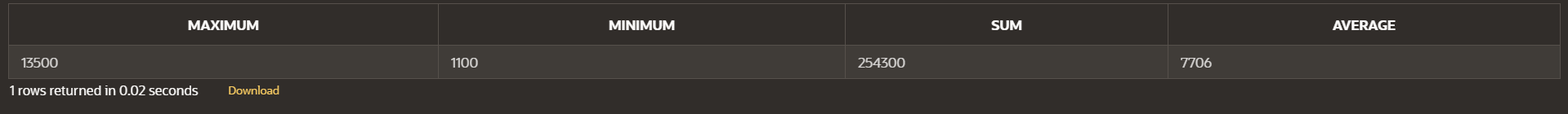
**FindtheSolutionforthefollowing:**

Determinethevalidityofthefollowingthreestatements.CircleeitherTrueorFalse.

1. Groupfunctionsworkacrossmanyrowstoproduceoneresultpergroup.True/False-TRUE
2. Group functions include nulls in calculations.True/False- FALSE
3. TheWHEREclauserestrictsrowspriortoinclusioninagroupcalculation.True/False-FALSE
4. Find the highest, lowest, sum, and average salary of all employees. Label thecolumnsMaximum,Minimum,Sum,andAverage,respectively.Roundyourresultstothenearestwhole number

SELECTROUND(MAX(salary))ASMaximum,ROUND(MIN(salary))ASMinimum,ROUND(SUM(salary))ASSum,ROUND(AVG(salary)) ASAverage

FROMemployees;



1. Modifytheabovequerytodisplaytheminimum,maximum,sum,andaveragesalaryforeachjobtype.

SELECTROUND(MAX(salary))ASMaximum,ROUND(MIN(salary))ASMinimum,ROUND(SUM(salary))ASSum,ROUND(AVG(salary)) ASAverage

FROM employeesjoindepartment

ondepartment.dept\_id=employees.department\_idgroupbydept\_name;

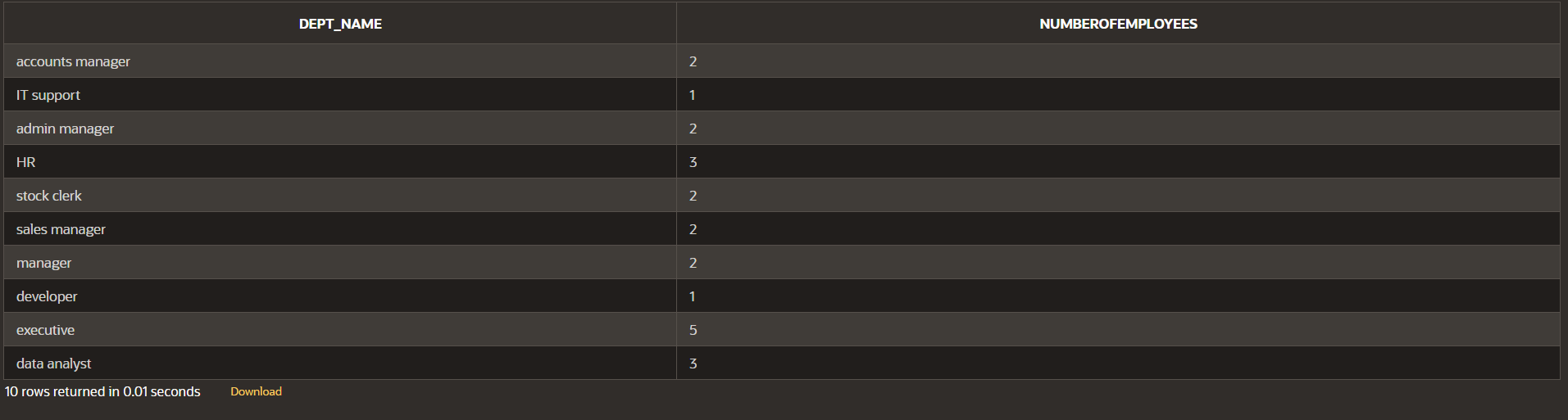


1. Writeaquerytodisplaythenumberofpeoplewiththesamejob.Generalizethequerysothat

theuserintheHRdepartmentispromptedforajobtitle.

SELECTd.dept\_name,COUNT(\*)ASNumberOfEmployeesFROMEmployeese

joindepartmentdone.department\_id=d.dept\_idgroupbyd.dept\_name;

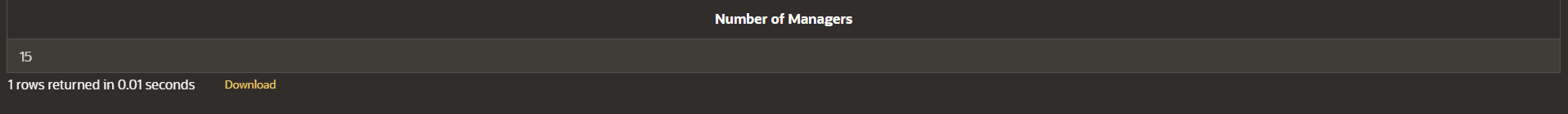


1. Determinethenumberofmanagerswithoutlistingthem.LabelthecolumnNumberofManagers

SELECTCOUNT(DISTINCTMANAGER\_ID)AS"NumberofManagers"

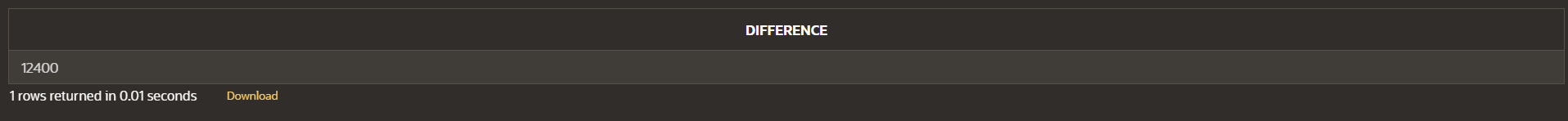
FROMEmployees

WHEREMANAGER\_IDISNOTNULL;



1. Findthedifferencebetweenthehighestandlowestsalaries.LabelthecolumnDIFFERENCE.

selectmax(salary)-min(salary)as"DIFFERENCE"fromemployees;



1. Create a report to display the manager number and the salary of the lowest-paidemployeeforthatmanager.Excludeanyonewhosemanagerisnotknown.Excludeanygroups where the minimum salary is $6,000 or less. Sort the output in descending orderofsalary.

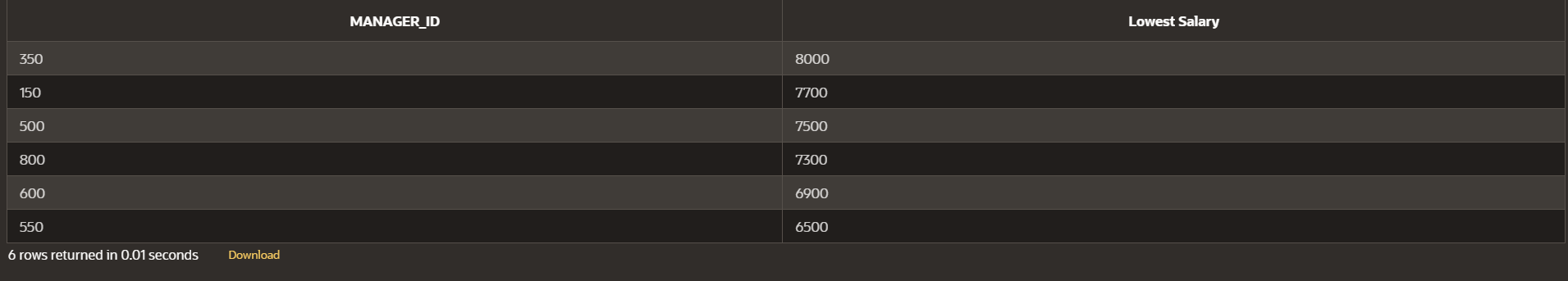
SELECTMANAGER\_ID,MIN(SALARY)AS"LowestSalary"

FROMEmployees

WHEREMANAGER\_IDISNOTNULLGROUPBYMANAGER\_ID

HAVINGMIN(SALARY)>6000

ORDERBY"LowestSalary"DESC;



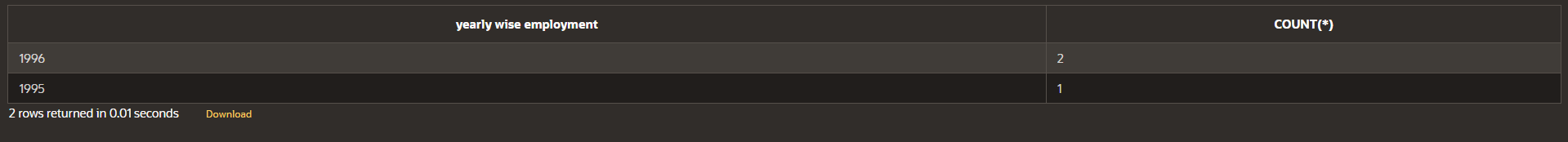
1. Createaquerytodisplaythetotalnumberofemployeesand,ofthattotal,thenumberofemployeeshiredin1995,1996,1997,and1998.Createappropriate

columnheadings.

SELECTEXTRACT(YEARFROMhire\_date)AS"yearlywiseemployment",COUNT(\*)FROMemployees

GROUPBYEXTRACT(YEARFROMhire\_date)

HAVINGEXTRACT(YEARFROMhire\_date)IN(1995,1996,1997,1998);



1. Create a matrix query to display the job, the salary for that job based on departmentnumber,andthetotalsalaryforthatjob,fordepartments20,50,80,and90,givingeachcolumnanappropriateheading.

selectd.dept\_name,sum(e.salary)fromemployees e

joindepartmentdone.department\_id=d.dept\_idwheredepartment\_idin(20,50,80,90)

groupbyd.dept\_name;



1. Writeaquerytodisplayeachdepartment‘sname,location,numberofemployees,andthe

averagesalaryforalltheemployeesinthatdepartment.Labelthecolumnname-Location,

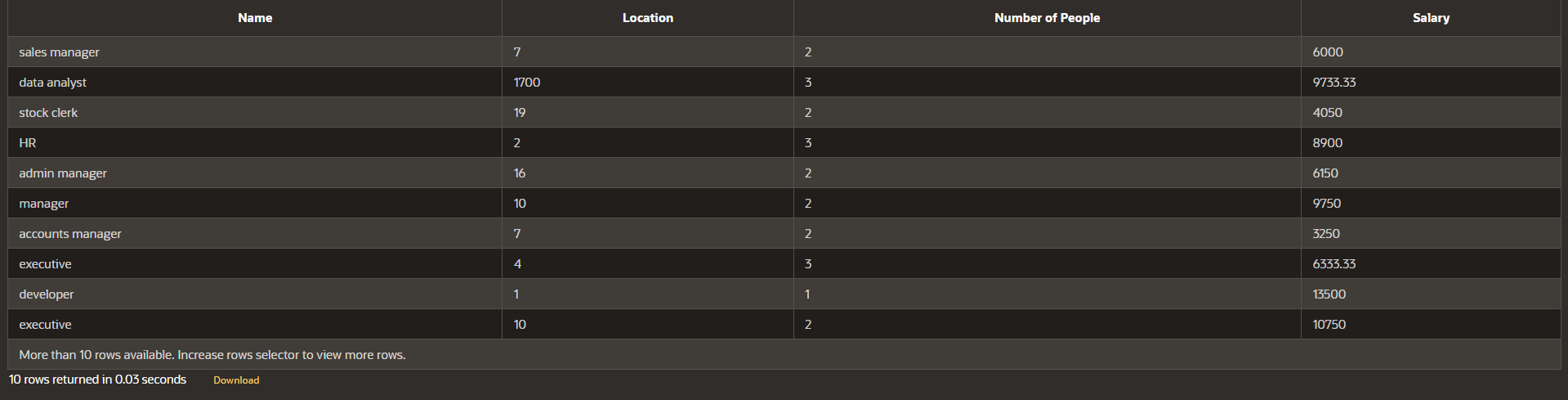
Numberofpeople,andsalaryrespectively.Roundtheaveragesalarytotwodecimalplaces.

SELECT d.dept\_name AS "Name", d.Location\_id AS "Location",COUNT(e.department\_id)AS"NumberofPeople",ROUND(AVG(e.Salary),2)AS"Salary"

FROMdepartmentd

JOINemployeeseONd.dept\_id=e.department\_id

GROUPBYd.dept\_name,d.location\_id;



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

**PROGRAM1**

**WriteaPL/SQLblocktocalculatetheincentiveofanemployeewhoseIDis110.**

DECLARE

pl\_emp\_id employees.employee\_id%TYPE := 110;pl\_salaryemployees.salary%TYPE;

pl\_incentive NUMBER;BEGIN

SELECT salary INTO pl\_salaryFROMemployees

WHEREemployee\_id=pl\_emp\_id;

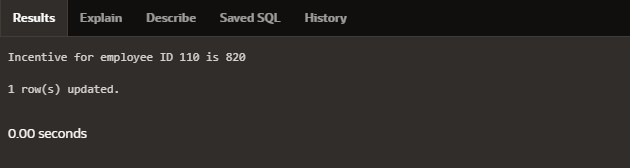
pl\_incentive:=pl\_salary\*0.10;UPDATEemployees

SET incentive = pl\_incentiveWHEREemployee\_id=pl\_emp\_id;

DBMS\_OUTPUT.PUT\_LINE('Incentive for employee ID ' || pl\_emp\_id || ' is ' ||pl\_incentive);

COMMIT;

END;



**PROGRAM2**

**WriteaPL/SQLblocktoshowaninvalidcase-insensitivereferencetoaquotedandwithoutquoted user-definedidentifier**.

DECLARE

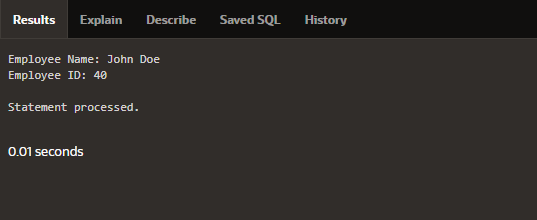
employeeNameVARCHAR2(100);"EmployeeID"NUMBER;

BEGIN

employeeName := 'John Doe';"EmployeeID":=40;

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || employeeName);DBMS\_OUTPUT.PUT\_LINE('Employee ID:'||"EmployeeID");

END;



**PROGRAM3**

**WriteaPL/SQLblocktoadjustthesalaryoftheemployeewhoseID122.Sampletable:employees**

**DECLARE**

**v\_employee\_id NUMBER := 122;v\_salary NUMBER;v\_new\_salaryNUMBER;**

**v\_increase\_percentage NUMBER := 0.40;BEGIN**

**SELECT salary INTO v\_salaryFROMemployees**

**WHEREemployee\_id=v\_employee\_id;**

**v\_new\_salary := v\_salary + (v\_salary \* v\_increase\_percentage / 100);UPDATEemployees**

**SETsalary=v\_new\_salary**

**WHEREemployee\_id=v\_employee\_id;**

**DBMS\_OUTPUT.PUT\_LINE('Employee ID ' || v\_employee\_id || ' new salary: ' ||v\_new\_salary);**

**END;**

A screenshot of a computer

Description automatically generated

**PROGRAM4**

**WriteaPL/SQLblocktocreateaprocedureusingthe"IS[NOT]NULLOperator"andshow ANDoperatorreturnsTRUEifandonlyifbothoperandsareTRUE.**

**create or replace procedure check\_nullis**

**value1 number := 10;value2number:=null;**

**begin**

**if value1 is not null and value2 is null thendbms\_output.put\_line('Bothvaluesarenotnull!!');**

**else**

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

**end;**

**BEGIN**

**check\_null;END;**

A black screen with white text

Description automatically generated

**PROGRAM5**

**Write a PL/SQL block to describe the usage of LIKE operator including wildcardcharactersand**

**escapecharacter.**

**declare**

**v\_employeename employees.first\_name%type;v\_employeeidNUMBER :=122;**

**begin**

**selectfirst\_nameintov\_employeenamefromemployees**

**where first\_name like '%e%' and employee\_id = v\_employeeid;DBMS\_OUTPUT.PUT\_LINE(v\_employeename);**

**END;**

**PROGRAM6**

**Write a PL/SQL program to arrange the number of two variable in such a way that thesmallnumberwillstoreinnum\_smallvariableandlargenumberwillstoreinnum\_largevariable.**

**declare**

**abnumber:=10;**

**cd number :=20;num\_small number;num\_large number;begin**

**if ab>cd thennum\_small :=cd;num\_large :=ab;else**

**num\_small :=ab;num\_large :=cd;endif;**

**dbms\_output.put\_line('small number = '||num\_small);dbms\_output.put\_line('large number = '||num\_large);End;**

A screenshot of a computer

Description automatically generated

**PROGRAM7**

**Write a PL/SQL procedure to calculate the incentive on a target achieved and display themessageeithertherecordupdated ornot.**

**create or replace procedure calculate\_incentive(p\_emp\_idemployees.employee\_id%type,p\_targetnumber)**

**is**

**v\_incentivenumber(7,2);v\_salaryemployees.salary%type;**

**begin**

**select salary into v\_salaryfromemployees**

**whereemployee\_id=p\_emp\_id;**

**if p\_target >= 100000 thenv\_incentive:=v\_salary\*0.1;**

**dbms\_output.put\_line('Incentive of ' || v\_incentive || ' calculated for employee ID ' ||p\_emp\_id);**

**else**

**dbms\_output.put\_line('No incentive for employee ID ' || p\_emp\_id);end if;**

**End;**

A black screen with white text

Description automatically generated

**PROGRAM8**

**Write a PL/SQL procedure to calculate incentive achieved according to the specific salelimit.**

**create or replace procedure incentive\_sale(p\_emp\_id employees.employee\_id%type,p\_salesnumber)**

**is**

**v\_incentive number(7,2);begin**

**if p\_sales > 100000 thenv\_incentive:=p\_sales\*0.1;**

**elsif p\_sales between 50000 and 100000 thenv\_incentive:=p\_sales \*0.05;**

**else**

**v\_incentive := 0;endif;**

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

**begin**

**incentive\_sale(122,500000);end;**

A black screen with white text

Description automatically generated

**PROGRAM9**

**Write a PL/SQL program to count number of employees in department 50 and checkwhether this department have any vacancies or not. There are 45 vacancies in thisdepartment.**

**declare**

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

**select count(\*) into no\_of\_emp from employees where department\_id=50;ifno\_of\_emp<vacancies then**

**dbms\_output.put\_line('vacancies are available');else**

**dbms\_output.put\_line('vacancies are not available');end if;**

**end;**

A screen shot of a computer

Description automatically generated

**PROGRAM10**

**Write a PL/SQL program to count number of employees in a specific department andcheck whether this department have any vacancies or not. If any vacancies, how manyvacanciesareinthatdepartment.**

**declare**

**v\_department\_id number := 55;v\_emp\_count number;v\_vacancies number:=50;**

**begin**

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

**wheredepartment\_id=v\_department\_id;**

**ifv\_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;**

A screenshot of a computer

Description automatically generated

**PROGRAM11**

**WriteaPL/SQLprogramtodisplaytheemployeeIDs,names,jobtitles,hiredates,andsalariesofallemployees.**

**begin**

**for i in (select employee\_id, first\_name || ' ' || last\_name as name, job\_id, hire\_date,salaryfromemployees)**

**loop**

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

**||',HireDate:'||i.hire\_date||',Salary:'||i.salary);endloop;**

**end;**

A computer screen with white text

Description automatically generated

**PROGRAM12**

**Write a PL/SQL program to display the employee IDs, names, and department names ofallemployees.**

**begin**

**foriin(selecte.employee\_id,e.first\_name||''||e.last\_nameasname,d.dept\_namefromemployees e**

**join department d on e.employee\_id = d.dept\_id) loopdbms\_output.put\_line('ID:'||i.employee\_id||',Name:'||i.name||',Department:'||**

**i.dept\_name);endloop;**

**End;**

A computer screen with white text

Description automatically generated

**PROGRAM13**

**WriteaPL/SQLprogramtodisplaythejobIDs,titles,andminimumsalariesofalljobs.**

**begin**

**for rec in (select e.employee\_id, d.dept\_name, min(salary) as min\_salary fromemployees**

**ejoindepartmentd**

**one.employee\_ID=d.dept\_id**

**group by e.employee\_id , d.dept\_name)loop**

**dbms\_output.put\_line('Job ID: ' || rec.employee\_id || ', Title: ' || rec.dept\_name || ',MinSalary:'||rec.min\_salary);**

**end loop;End;**

A black rectangle with white text

Description automatically generated

**WriteaPL/SQLprogramtodisplaythejobIDs,titles,andminimumsalariesofalljobs.**

**begin**

**for rec in (select e.employee\_id, d.dept\_name, min(salary) as min\_salary fromemployees**

**ejoindepartmentd**

**one.employee\_ID=d.dept\_id**

**group by e.employee\_id , d.dept\_name)loop**

**dbms\_output.put\_line('Job ID: ' || rec.employee\_id || ', Title: ' || rec.dept\_name || ',MinSalary:'||rec.min\_salary);**

**end loop;End;**

A black rectangle with white text

Description automatically generated

**PROGRAM14**

**WriteaPL/SQLprogramtodisplaytheemployeeIDs,names,andjobhistorystartdatesofall**

**Employees.**

**Begin**

**for rec in (select employee\_id, first\_name || ' ' || last\_name as name, hire\_datefromemployees) loop**

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

**|| rec.hire\_date);endloop;**

**end;**

A black screen with white text

Description automatically generated

**PROGRAM15**

**Write a PL/SQL program to display the employee IDs, names, and job history end datesofall employees.**

**BEGIN**

**FOR rec IN (SELECT employee\_id, first\_name || ' ' || last\_name AS name, end\_dateFROMemployees)**

**LOOP**

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

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

**NVL(TO\_CHAR(rec.end\_date,'YYYY-MM-DD'),'StillActive'));**

**END LOOP;END;**

A black screen with white text

Description automatically generated

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

**Program1**

**FACTORIALOFANUMBERUSINGFUNCTION**

**DECLARE**

**nNUMBER:=10;**

**resultNUMBER;**

**FUNCTION itfact(num NUMBER) RETURN NUMBER ISfactNUMBER :=1;**

**BEGIN**

**FORiIN1..numLOOP**

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

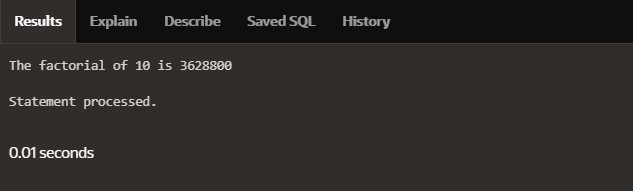
**ENDLOOP;**

**RETURN fact;END;**

**BEGIN**

**result:=itfact(n);**

**DBMS\_OUTPUT.PUT\_LINE('The factorial of ' || n || ' is ' || result);END;**



**Program2**

**Write a PL/SQL program using Procedures IN,INOUT,OUT parameters to retrieve thecorrespondingbook informationinlibrary**

**CREATEORREPLACEPROCEDUREbook\_info(**

**p\_book\_id IN NUMBER,p\_author OUT VARCHAR2,p\_title OUT VARCHAR2,p\_published\_dateOUTDATE**

**) ASBEGIN**

**SELECT author, title, published\_dateINTO p\_author, p\_title, p\_published\_dateFROMbooks**

**WHEREbook\_id=p\_book\_id;**

**EXCEPTION**

**WHENNO\_DATA\_FOUNDTHEN**

**p\_author := NULL;p\_title:=NULL;**

**p\_published\_date := NULL;WHENOTHERS THEN**

**RAISE;**

**ENDbook\_info;**

**DECLARE**

**v\_author VARCHAR2(100);v\_title VARCHAR2(100);v\_published\_date DATE;v\_book\_idNUMBER:=1;**

**BEGIN**

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

**IF v\_author IS NOT NULL THENDBMS\_OUTPUT.PUT\_LINE('Book ID: ' || v\_book\_id);DBMS\_OUTPUT.PUT\_LINE('Author: ' || v\_author);DBMS\_OUTPUT.PUT\_LINE('Title:'||v\_title);**

**DBMS\_OUTPUT.PUT\_LINE('Published Date: ' || TO\_CHAR(v\_published\_date, 'YYYY-MM-DD'));**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE('No book found with ID: ' || v\_book\_id);END IF;**

**END;**



|  |  |  |
| --- | --- | --- |
| **Ex.No.: 13** | | **WORKING WITH TRIGGERS** |
| **Date:** | 20/09/2024 |

**Program1**

**WriteacodeinPL/SQLtodevelopatriggerthatenforcesreferentialintegritybypreventingthedeletionofaparentrecordifchildrecords exist.**

**CREATE OR REPLACE TRIGGER prevent\_parent\_deletionBEFOREDELETE ONemployees**

**FOR EACH ROWDECLARE**

**pl\_dept\_count NUMBER;BEGIN**

**SELECTCOUNT(\*)**

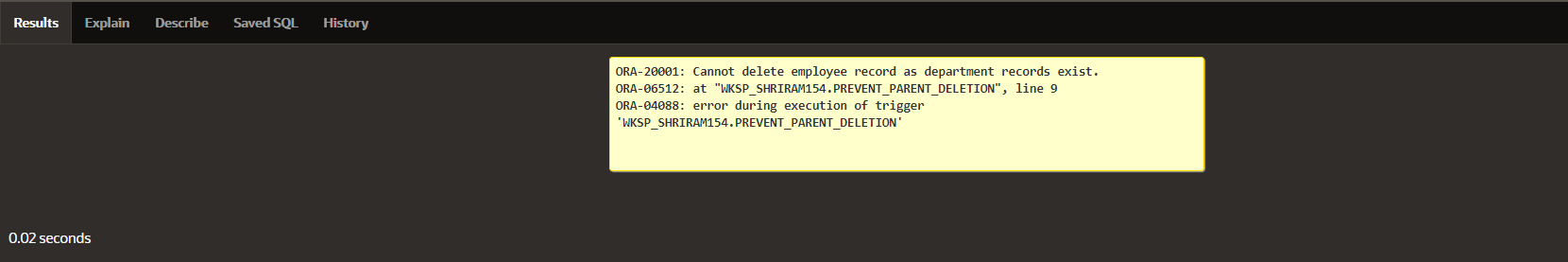
**INTO pl\_dept\_countFROMdepartment**

**WHERE dept\_id = :OLD.employee\_id;IFpl\_dept\_count >0THEN**

**RAISE\_APPLICATION\_ERROR(-20001, 'Cannot delete employee record asdepartmentrecords exist.');**

**END IF;END;**

**DELETE FROM employeesWHEREemployee\_id=70;**



**Program2**

**WriteacodeinPL/SQLtocreateatriggerthatchecksforduplicatevaluesinaspecificcolumnand raisesanexception iffound.**

**CREATE OR REPLACE TRIGGER prevent\_duplicate\_manager\_idBEFOREINSERTORUPDATE ONemployees**

**FOR EACH ROWDECLARE**

**pl\_count NUMBER;BEGIN**

**SELECTCOUNT(\*)**

**INTO pl\_countFROMemployees**

**WHERE manager\_id = :NEW.manager\_idAND employee\_id != :NEW.employee\_id;IFpl\_count>0THEN**

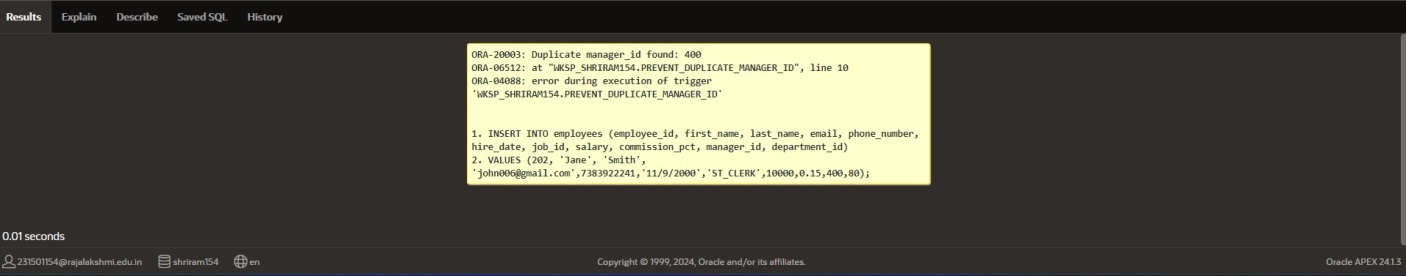
**RAISE\_APPLICATION\_ERROR(-20003,'Duplicatemanager\_idfound:'||**

**:NEW.manager\_id);ENDIF;**

**END;**

**INSERT INTO employees (employee\_id, first\_name, last\_name, email, phone\_number,hire\_date,job\_id,salary,commission\_pct,manager\_id,department\_id)**

**VALUES (202, 'Jane', 'Smith','john006@gmail.com',7383922241,'11/9/2000','ST\_CLERK',10000,0.15,400,80);**



**Program3**

**WriteacodeinPL/SQLtocreateatriggerthatrestrictstheinsertionofnewrowsifthetotalofa**

**column'svaluesexceedsacertainthreshold.**

**CREATE OR REPLACE TRIGGER restrict\_salary\_insertionBEFOREINSERTONemployees**

**FOR EACH ROWDECLARE**

**total\_salary NUMBER;thresholdNUMBER:=100000;**

**BEGIN**

**SELECT SUM(salary)INTO total\_salaryFROMemployees;**

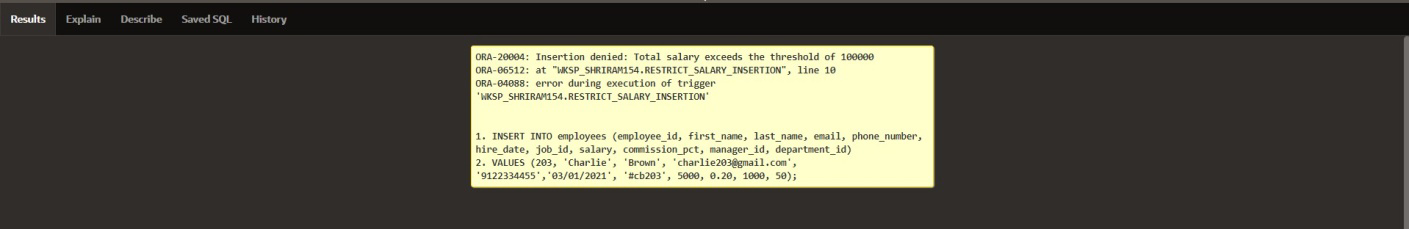
**IF(total\_salary+:NEW.salary)>thresholdTHEN**

**RAISE\_APPLICATION\_ERROR(-20004, 'Insertion denied: Total salary exceeds thethresholdof'||threshold);**

**END IF;END;**

**INSERT INTO employees (employee\_id, first\_name, last\_name, email, phone\_number,hire\_date,job\_id,salary,commission\_pct,manager\_id,department\_id)**

**VALUES (203, 'Charlie', 'Brown', 'charlie203@gmail.com', '9122334455','03/01/2021','#cb203',5000,0.20,1000,50);**



**PROGRAM 4**

**Write a code in PL/SQL to design a trigger that captures changes made to specificcolumnsand logstheminanaudittable.**

**CREATE OR REPLACE TRIGGER audit\_changesAFTER UPDATE OF salary, job\_id ON employeesFOREACH ROW**

**BEGIN**

**IF:OLD.salary!=:NEW.salaryOR:OLD.job\_id!=:NEW.job\_idTHENINSERTINTOemployee\_audit (**

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

**)VALUES (**

**:OLD.employee\_id,**

**:OLD.salary,**

**:NEW.salary,**

**:OLD.job\_id,**

**:NEW.job\_id,SYSTIMESTAMP,USER**

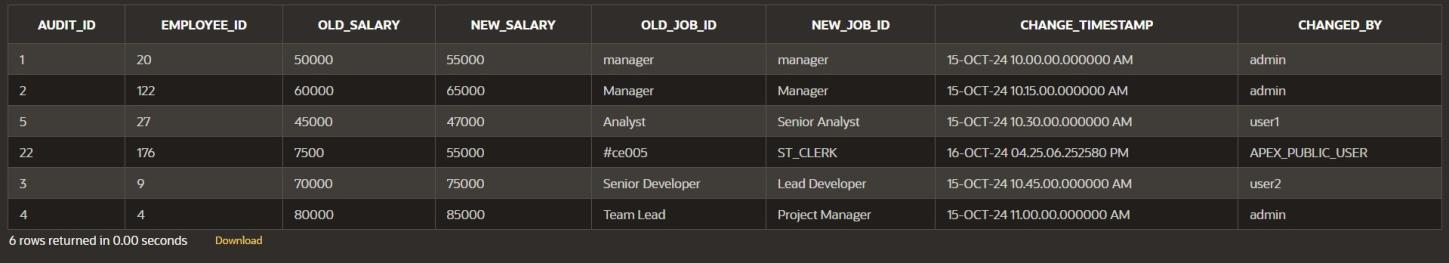
**);ENDIF;**

**END;**

**UPDATEemployees**

**SET salary = 55000, job\_id = 'ST\_CLERK'WHEREemployee\_id =176;**

**SELECT\*FROMemployee\_audit;**



**PROGRAM 5**

**WriteacodeinPL/SQLtoimplementatriggerthatrecordsuseractivity(inserts,updates,deletes)inanauditlogfora givensetoftables.**

**CREATE OR REPLACE TRIGGER trg\_audit\_employeesAFTER INSERT OR UPDATE OR DELETE ON employeesFOREACH ROW**

**DECLARE**

**v\_old\_values CLOB;v\_new\_valuesCLOB;**

**BEGIN**

**IFINSERTINGTHEN**

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

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

**'salary:'||:NEW.salary;**

**INSERT INTO audit\_log (action, table\_name, record\_id, changed\_by, new\_values)VALUES('INSERT','employees',:NEW.employee\_id,USER,v\_new\_values);**

**ELSIFUPDATINGTHEN**

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

**'salary:'||:OLD.salary;**

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

**'salary:'||:NEW.salary;**

**INSERT INTO audit\_log (action, table\_name, record\_id, changed\_by, old\_values,new\_values)**

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

**ELSIFDELETINGTHEN**

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

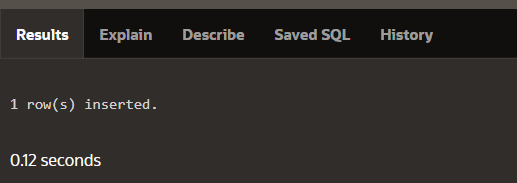
**'salary: ' || :OLD.salary;v\_new\_values:=NULL;**

**INSERT INTO audit\_log (action, table\_name, record\_id, changed\_by, old\_values)VALUES('DELETE','employees',:OLD.employee\_id,USER,v\_old\_values);**

**ENDIF;**

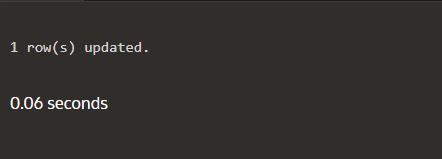
**ENDtrg\_audit\_employees;**

**INSERT INTO employees (employee\_id, first\_name, salary)VALUES(3,'Ball', 50000);**



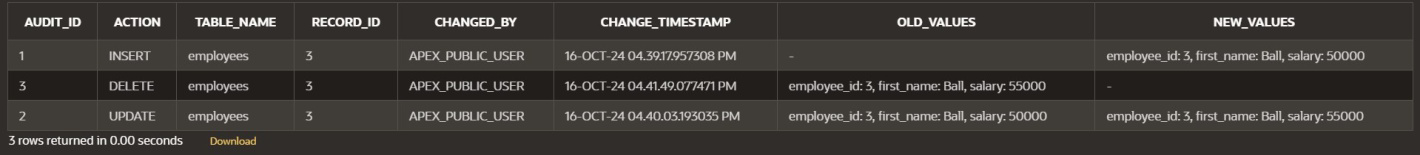
**UPDATE employeesSETsalary=55000**

**WHEREemployee\_id=3;**



**DELETE FROM employeesWHEREemployee\_id=3;**

**SELECT\*FROMaudit\_log;**



**PROGRAM 6**

**WriteacodeinPL/SQLtoimplementatriggerthatautomaticallycalculatesandupdatesarunning totalcolumnforatablewhenever new rowsareinserted.**

**CREATE TABLE transactions (transaction\_id NUMBER PRIMARY KEY,amountNUMBER,**

**running\_totalNUMBER**

**);**

**CREATE OR REPLACE TRIGGER update\_running\_totalFORINSERT ONtransactions**

**COMPOUNDTRIGGER**

**TYPEamount\_arrayISTABLEOFNUMBERINDEXBYPLS\_INTEGER;**

**new\_amountsamount\_array;**

**BEFORE EACH ROW ISBEGIN**

**new\_amounts(:NEW.transaction\_id) := :NEW.amount;ENDBEFORE EACH ROW;**

**AFTER STATEMENT ISBEGIN**

**DECLARE**

**v\_total NUMBER;BEGIN**

**SELECT NVL(MAX(running\_total), 0)INTOv\_total**

**FROMtransactions;**

**FOR i IN new\_amounts.FIRST .. new\_amounts.LAST LOOPv\_total:=v\_total+new\_amounts(i);**

**UPDATEtransactions**

**SET running\_total = v\_totalWHEREtransaction\_id=i;**

**END LOOP;END;**

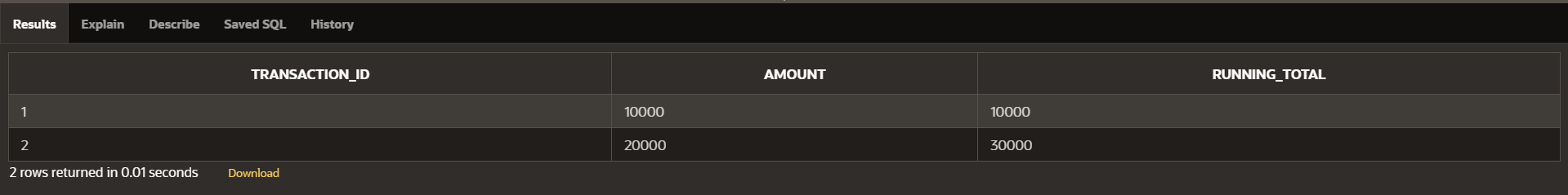
**ENDAFTERSTATEMENT;**

**ENDupdate\_running\_total;**

**INSERTINTOtransactions(transaction\_id,amount)**

**VALUES(1,10000);**

**INSERT INTO transactions (transaction\_id, amount)VALUES(2,20000);**



**PROGRAM 7**

**WriteacodeinPL/SQLtocreateatriggerthatvalidatestheavailabilityofitemsbeforeallowinganorder tobeplaced,consideringstocklevelsand pendingorders.**

**CREATETABLEinventory(item\_idNUMBERPRIMARYKEY,**

**item\_name VARCHAR2(100),stock\_levelNUMBER**

**);**

**CREATETABLEorders(**

**order\_id NUMBER PRIMARY KEY,item\_idNUMBER,**

**quantity NUMBER,order\_statusVARCHAR2(20),**

**CONSTRAINTfk\_itemFOREIGNKEY(item\_id)REFERENCESinventory(item\_id)**

**);**

**CREATE OR REPLACE TRIGGER validate\_stock\_before\_orderBEFOREINSERTONorders**

**FOR EACH ROWDECLARE**

**v\_stock\_level NUMBER;v\_pending\_ordersNUMBER;**

**BEGIN**

**SELECT stock\_levelINTO v\_stock\_levelFROMinventory**

**WHERE item\_id = :NEW.item\_id;SELECT NVL(SUM(quantity), 0)INTOv\_pending\_orders**

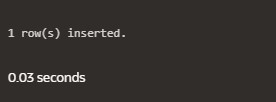
**FROMorders**

**WHERE item\_id = :NEW.item\_idANDorder\_status='Pending';**

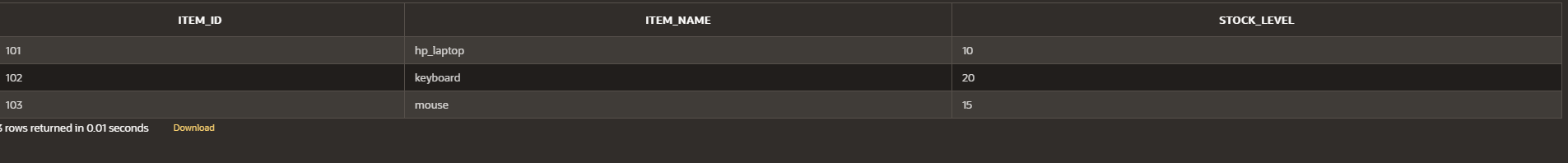
**IF (:NEW.quantity + v\_pending\_orders) > v\_stock\_level THENRAISE\_APPLICATION\_ERROR(-20001,'Insufficientstockforitem:'||:NEW.item\_id);**

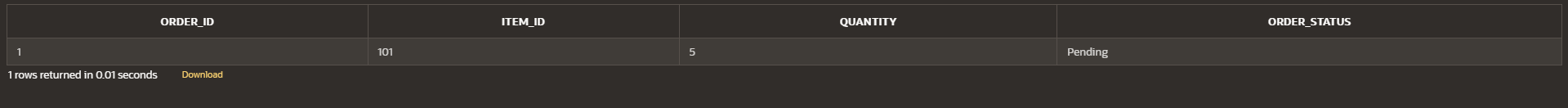
**END IF;END;**

**INSERTINTOorders(order\_id,item\_id,quantity,order\_status)VALUES(1,101, 5,'Pending');**



**INSERTINTOorders(order\_id,item\_id,quantity,order\_status)VALUES(2,103, 20, 'Pending');**





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

1. **WriteaMongoDBquerytofindtherestaurantId,name,boroughandcuisineforthoserestaurants which prepared dish except 'American' and 'Chinees' or restaurant's namebeginswithletter'Wil'.**

**db.restaurants.find(**

**{**

**$or:[**

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

**{name:{$regex:/^Wil/i}}**

**]**

**},**

**{**

**restaurant\_id:1,**

**name:1,**

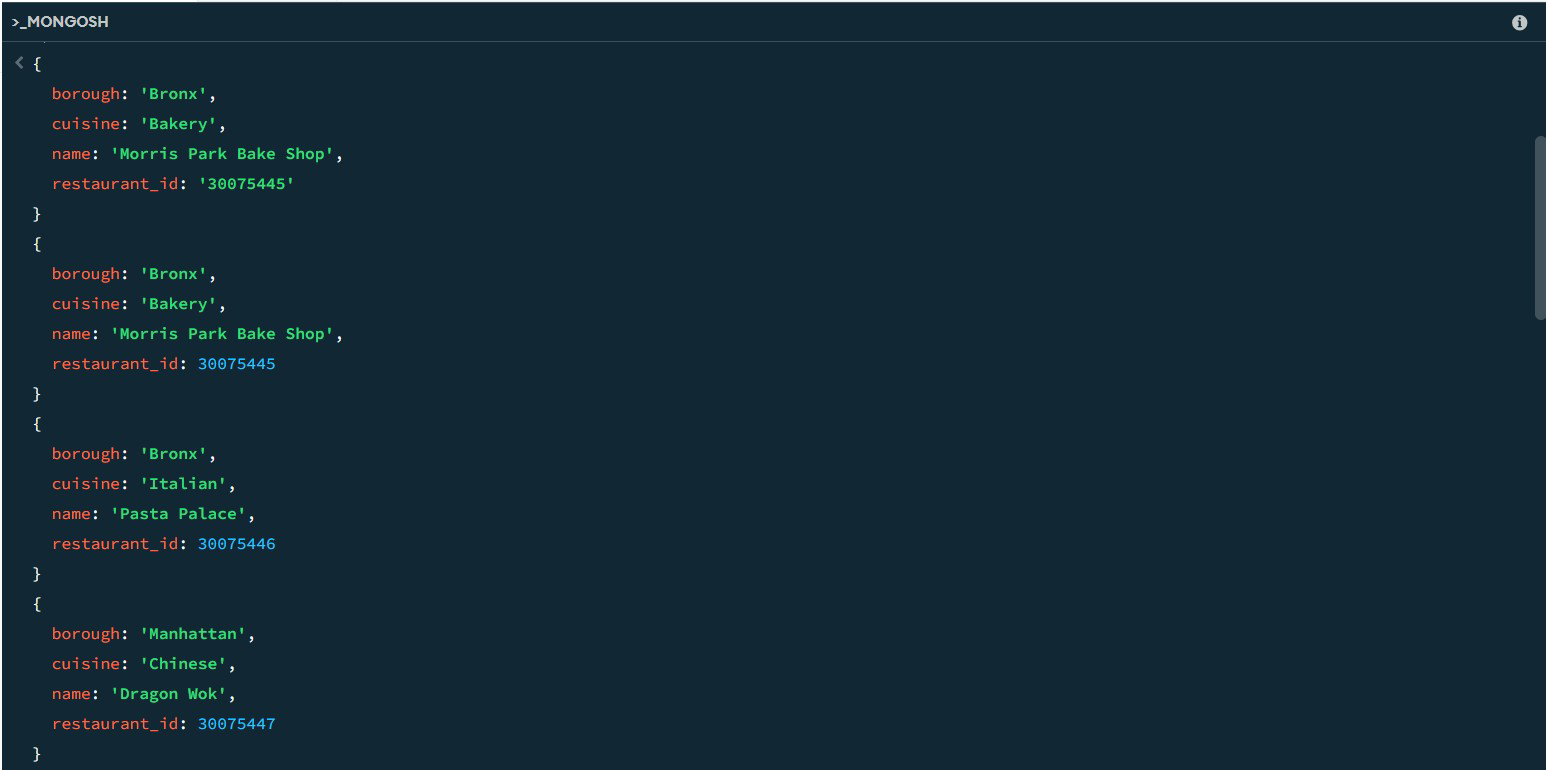
**borough:1,**

**cuisine:1,**

**\_id:0**

**}**

**);**



1. **Write a MongoDB query to find the restaurant Id, name, and grades for thoserestaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z"among manyofsurvey dates..**

**db.restaurants.find(**

**{**

**grades:{**

**$elemMatch: {grade: "A",score:11**

**}**

**}**

**},**

**{**

**restaurant\_id:1,**

**name:1,**

**grades:1,**

**\_id:0**

**}**

**);**



1. **Write a MongoDB query to find the restaurant Id, name and grades for thoserestaurants where the 2nd element of grades array contains a grade of "A" and score 9onanISODate"2014-08-11T00:00:00Z".**

**db.restaurants.find(**

**{**

**"grades.1":{**

**$elemMatch: {grade: "A",score:9**

**}**

**}**

**},**

**{**

**restaurant\_id:1,**

**name:1,**

**grades:1,**

**\_id:0**

**}**

**);**

1. **Write a MongoDB query to find the restaurant Id, name, address and geographicallocation for those restaurants where 2nd element of coord array contains a value whichismorethan42andupto52..**

**db.restaurants.find(**

**{**

**"address.coord.1":{$gt:42,$lte:52}**

**},**

**{**

**restaurant\_id:1,**

**name:1,**

**address:1,**

**\_id:0**

**}**

**);**

1. **Write a MongoDB query to arrange the name of the restaurants in ascending orderalongwithallthecolumns.**

**db.restaurants.find().sort({name:1});SAMPLEOUTPUT:-**

**{**

**\_id:ObjectId('671b5e6d56ec9972ca8f5dc4'),address:{**

**building: 5566,coord:[**

**-73.867377,**

**40.854047**

**],**

**street: '28th Avenue',zipcode:10490**

**},**

**borough: 'Bronx',cuisine: 'BBQ',grades:[**

**{**

**date:2014-03-03T00:00:00.028Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2013-09-11T00:00:00.028Z,**

**grade: 'A',score:7**

**},**

**{**

**date:2013-01-24T00:00:00.028Z,**

**grade: 'A',score:11**

**},**

**{**

**date:2011-11-23T00:00:00.028Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.028Z,**

**grade:'B',**

**score:15**

**}**

**],**

**name: 'BBQ Haven',restaurant\_id:30075473**

**}**

**{**

**\_id:ObjectId('671b5dab56ec9972ca8f5db0'),address:{**

**building: 5566,coord:[**

**-73.859377,**

**40.850047**

**],**

**street: '8th Avenue',zipcode:10470**

**},**

**borough: 'Manhattan',cuisine: 'French',grades:[**

**{**

**date:2014-03-03T00:00:00.008Z,**

**grade: 'A',score:7**

**},**

**{**

**date:2013-09-11T00:00:00.008Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2013-01-24T00:00:00.008Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.008Z,**

**grade: 'B',score:15**

**},**

**{**

**date:2011-03-10T00:00:00.008Z,**

**grade: 'A',score:6**

**}**

**],**

**name: 'Bistro Belle',restaurant\_id:30075453**

**}**

1. **Write a MongoDB query to arrange the name of the restaurants in descending alongwithallthecolumns.**

**db.restaurants.find().sort({name:-1});**

**SAMPLEOUTPUT**

**{**

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

**building: 9900,coord:[**

**-73.868977,**

**40.854847**

**],**

**street: '32nd Avenue',zipcode:10494**

**},**

**borough: 'Manhattan',cuisine: 'Russian',grades:[**

**{**

**date:2014-03-03T00:00:00.032Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2013-09-11T00:00:00.032Z,**

**grade: 'B',score:5**

**},**

**{**

**date:2013-01-24T00:00:00.032Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-11-23T00:00:00.032Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2011-03-10T00:00:00.032Z,**

**grade: 'A',score:11**

**}**

**],**

**name: "Tsar's Table",restaurant\_id:30075477**

**}**

**{**

**\_id:ObjectId('671b5e6d56ec9972ca8f5dbe'),address:{**

**building: 9900,coord:[**

**-73.864977,**

**40.852847**

**],**

**street: '22nd Avenue',zipcode:10484**

**},**

**borough: 'Bronx',cuisine: 'Italian',grades:[**

**{**

**date:2014-03-03T00:00:00.022Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2013-09-11T00:00:00.022Z,**

**grade: 'B',score:5**

**},**

**{**

**date:2013-01-24T00:00:00.022Z,**

**grade: 'A',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.022Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.022Z,**

**grade: 'A',score:14**

**}**

**],**

**name: 'Trattoria Bella',restaurant\_id:30075467**

**}**

1. **WriteaMongoDBquerytoarrangethenameofthecuisineinascendingorderandforthatsamecuisine borough shouldbeindescending order.**

**db.restaurants.find().sort({ cuisine: 1, borough: -1 });SAMPLEOUTPUT:-**

**{**

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

**building: 2233,coord:[**

**-73.858177,**

**40.849447**

**],**

**street: '5th Avenue',zipcode:10467**

**},**

**borough: 'Bronx',cuisine:'American',**

**grades:[**

**{**

**date:2014-03-03T00:00:00.005Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2013-09-11T00:00:00.005Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.005Z,**

**grade: 'B',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.005Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.005Z,**

**grade: 'A',score:14**

**}**

**],**

**name: 'Burger Bistro',restaurant\_id:30075450**

**}**

**{**

**\_id:ObjectId('671b5e6d56ec9972ca8f5dc4'),address:{**

**building: 5566,coord:[**

**-73.867377,**

**40.854047**

**],**

**street: '28th Avenue',zipcode:10490**

**},**

**borough: 'Bronx',cuisine:'BBQ',**

**grades:[**

**{**

**date:2014-03-03T00:00:00.028Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2013-09-11T00:00:00.028Z,**

**grade: 'A',score:7**

**},**

**{**

**date:2013-01-24T00:00:00.028Z,**

**grade: 'A',score:11**

**},**

**{**

**date:2011-11-23T00:00:00.028Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.028Z,**

**grade: 'B',score:15**

**}**

**],**

**name: 'BBQ Haven',restaurant\_id:30075473**

**}**

1. **WriteaMongoDBquerytoknowwhetheralltheaddressescontainsthestreetornot.**

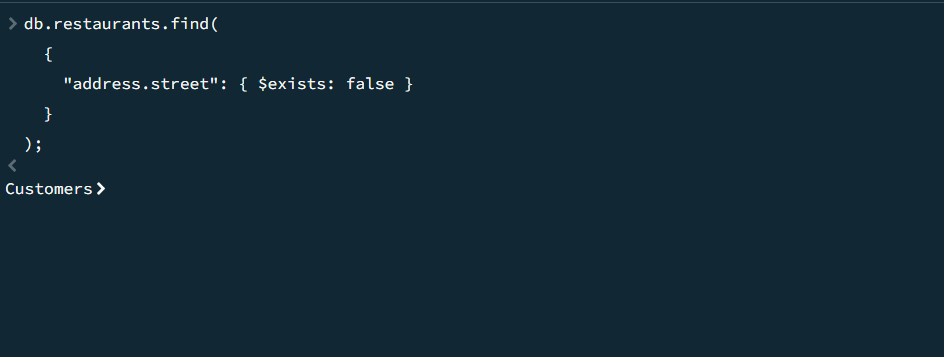
**db.restaurants.find(**

**{**

**"address.street":{$exists:false}**

**}**

**);**



1. **Write a MongoDB query which will select all documents in the restaurants collectionwherethecoord fieldvalue isDouble.**

**db.restaurants.find(**

**{**

**"address.coord":{$type:"double"}**

**}**

**);**

**SAMPLEOUTPUT:-**

**{**

**\_id:ObjectId('671b92d339ec8a9bc8b6588b'),address:{**

**building: '1007',coord:[**

**-73.856077,**

**40.848447**

**],**

**street: 'Morris Park Ave',zipcode:'10462'**

**},**

**borough: 'Bronx',cuisine: 'Bakery',grades:[**

**{**

**date:2014-03-03T00:00:00.000Z,**

**grade: 'A',score:2**

**},**

**{**

**date:2013-09-11T00:00:00.000Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.000Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.000Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.000Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Morris Park Bake Shop',restaurant\_id:'30075445'**

**}**

**{**

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

**building: 1234,coord:[**

**-73.856577,**

**40.848647**

**],**

**street: '1st Avenue',zipcode:10463**

**},**

**borough: 'Bronx',cuisine: 'Italian',grades:[**

**{**

**date:2014-03-03T00:00:00.001Z,**

**grade: 'A',score:5**

**},**

**{**

**date:2013-09-11T00:00:00.001Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2013-01-24T00:00:00.001Z,**

**grade: 'B',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.001Z,**

**grade: 'A',score:7**

**},**

**{**

**date:2011-03-10T00:00:00.001Z,**

**grade: 'A',score:15**

**}**

**],**

**name: 'Pasta Palace',restaurant\_id:30075446**

**}**

1. **Write a MongoDB query which will select the restaurant Id, name and grades forthose**

**restaurantswhichreturns0asaremainderafterdividingthescoreby7.**

**db.restaurants.find(**

**{**

**"grades.score":{$mod:[7,0]}**

**},**

**{**

**restaurant\_id:1,**

**name:1,**

**grades:1,**

**\_id:0**

**}**

**);**

**SAMPLEOUTPUT:-**

**{**

**grades:[**

**{**

**date:2014-03-03T00:00:00.000Z,**

**grade: 'A',score:2**

**},**

**{**

**date:2013-09-11T00:00:00.000Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.000Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.000Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.000Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Morris Park Bake Shop',restaurant\_id:'30075445'**

**}**

**{**

**grades:[**

**{**

**date:2014-03-03T00:00:00.001Z,**

**grade: 'A',score:5**

**},**

**{**

**date:2013-09-11T00:00:00.001Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2013-01-24T00:00:00.001Z,**

**grade: 'B',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.001Z,**

**grade: 'A',score:7**

**},**

**{**

**date:2011-03-10T00:00:00.001Z,**

**grade: 'A',score:15**

**}**

**],**

**name: 'Pasta Palace',restaurant\_id:30075446**

**}**

1. **Write a MongoDB query to find the restaurant name, borough, longitude and attitudeand cuisine for those restaurants which contains 'mon' as three letters somewhere in itsname.**

**db.restaurants.find(**

**{**

**name:{$regex:/mon/i}**

**},**

**{**

**name:1,**

**borough:1,**

**"address.coord.0": 1, // Longitude"address.coord.1": 1, // Latitudecuisine:1,**

**\_id:0**

**}**

**);**

1. **Write a MongoDB query to find the restaurant name, borough, longitude and latitudeandcuisineforthoserestaurantswhichcontain'Mad'asfirstthreelettersofitsname.**

**db.restaurants.find(**

**{**

**name:{$regex:/^Mad/i}**

**},**

**{**

**name:1,**

**borough:1,**

**"address.coord.0": 1, // Longitude"address.coord.1": 1, // Latitudecuisine:1,**

**\_id:0**

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have at least one grade with ascoreoflessthan5.**

**db.restaurants.find(**

**{**

**"grades.score":{$lt:5}**

**}**

**);**

**SAMPLEOUTPUT:-**

**{**

**\_id:ObjectId('671b92d339ec8a9bc8b6588b'),address:{**

**building:'1007',**

**coord:[**

**-73.856077,**

**40.848447**

**],**

**street: 'Morris Park Ave',zipcode:'10462'**

**},**

**borough: 'Bronx',cuisine: 'Bakery',grades:[**

**{**

**date:2014-03-03T00:00:00.000Z,**

**grade: 'A',score:2**

**},**

**{**

**date:2013-09-11T00:00:00.000Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.000Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.000Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.000Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Morris Park Bake Shop',restaurant\_id:'30075445'**

**}**

**{**

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

**building: 5678,coord:[**

**-73.856977,**

**40.848847**

**],**

**street: '2nd Avenue',zipcode:10464**

**},**

**borough: 'Manhattan',cuisine: 'Chinese',grades:[**

**{**

**date:2014-03-03T00:00:00.002Z,**

**grade: 'B',score:4**

**},**

**{**

**date:2013-09-11T00:00:00.002Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2013-01-24T00:00:00.002Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.002Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2011-03-10T00:00:00.002Z,**

**grade: 'B',score:16**

**}**

**],**

**name: 'Dragon Wok',restaurant\_id:30075447**

**}**

1. **Write a MongoDB query to find the restaurants that have at least one grade with ascoreoflessthan5and thatarelocatedintheborough ofManhattan.**

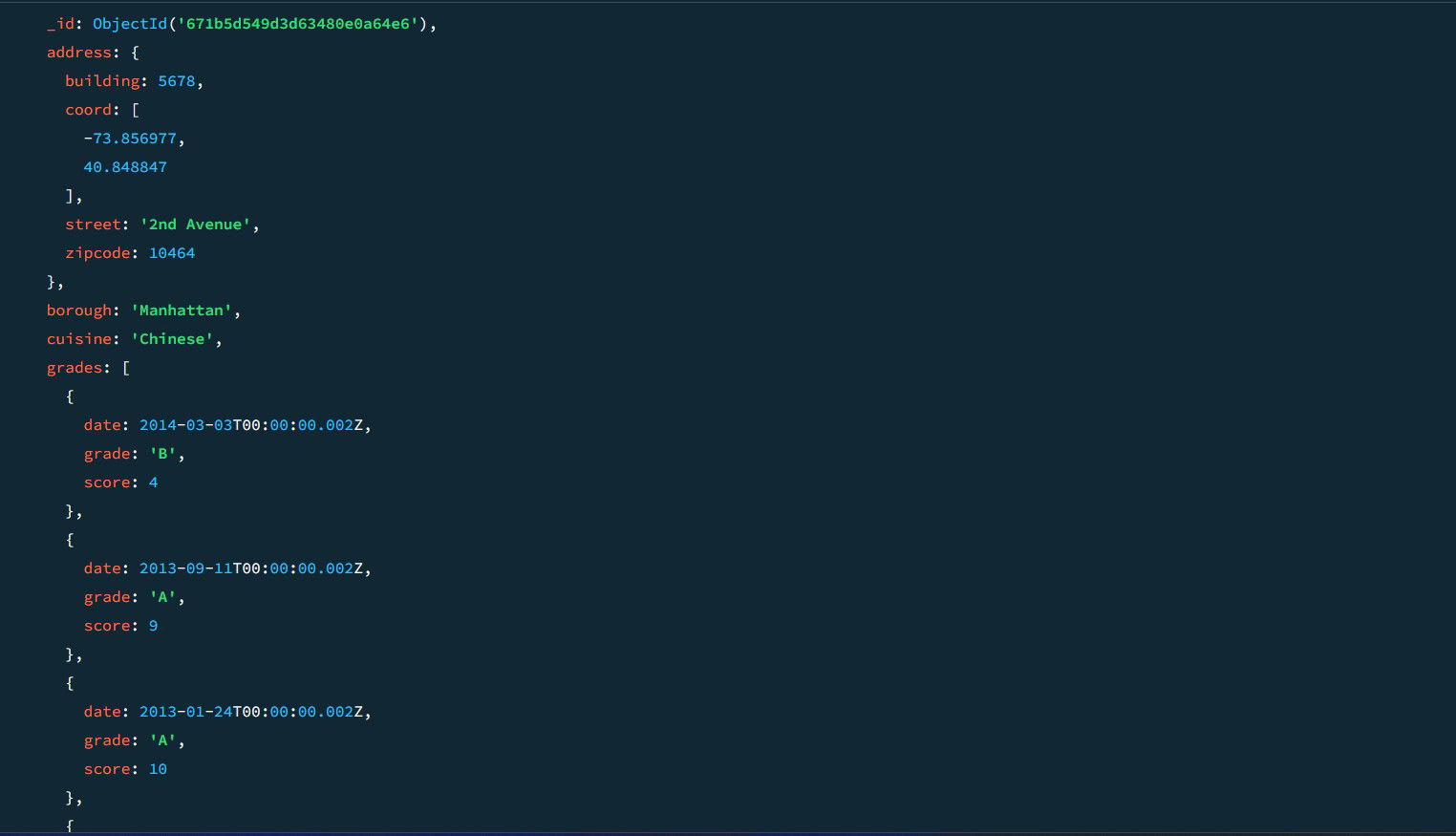
**db.restaurants.find(**

**{**

**"grades.score": { $lt: 5 },borough:"Manhattan"**

**}**

**);**



1. **Write a MongoDB query to find the restaurants that have at least one grade with ascoreoflessthan5andthatarelocatedintheboroughofManhattanorBrooklyn.**

**db.restaurants.find(**

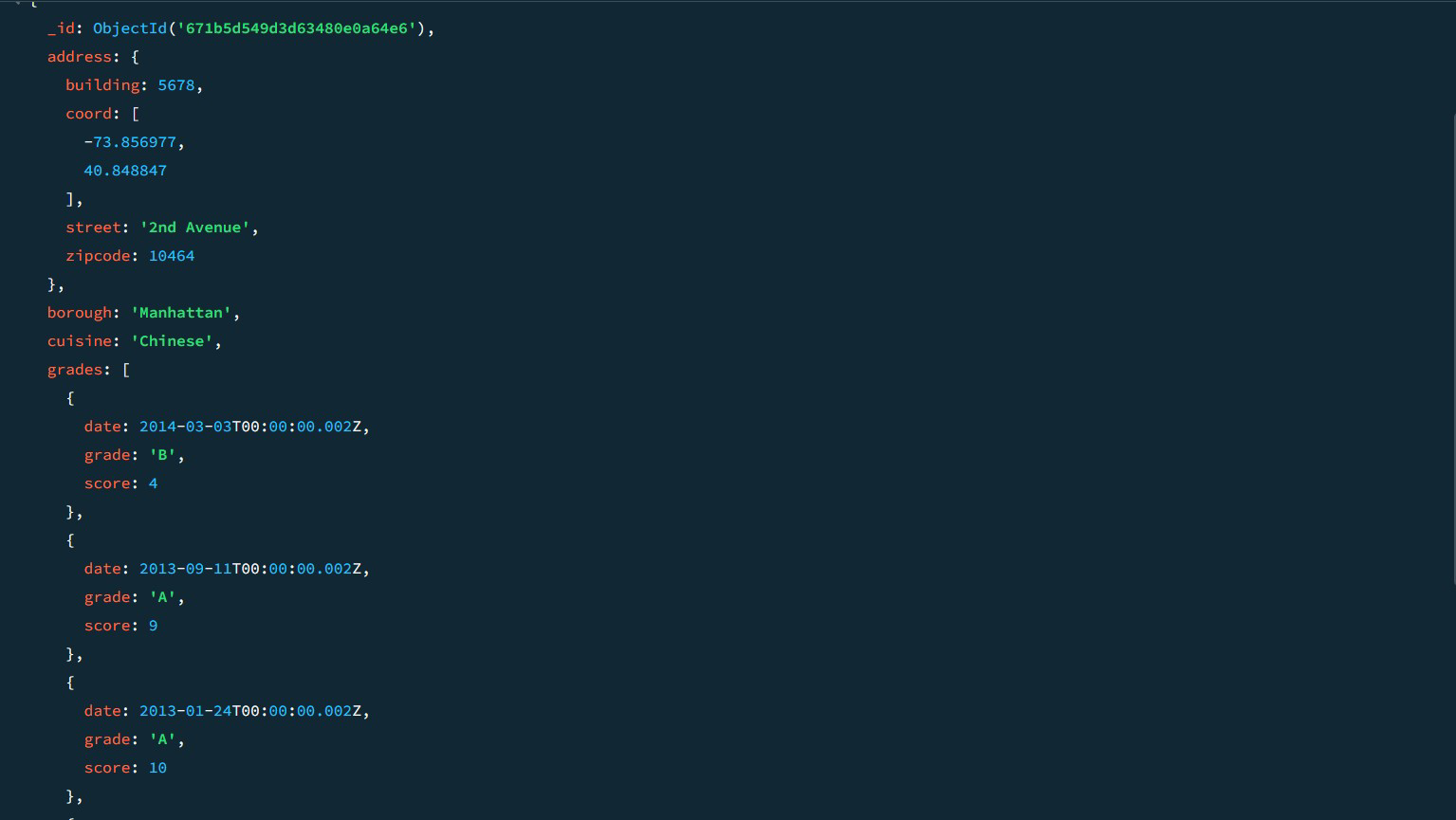
**{**

**"grades.score":{$lt:5},**

**borough:{$in:["Manhattan","Brooklyn"]}**

**}**

**);**



1. **Write a MongoDB query to find the restaurants that have at least one grade with ascoreoflessthan5andthatarelocatedintheboroughofManhattanorBrooklyn,andtheircuisine is notAmerican.**

**db.restaurants.find(**

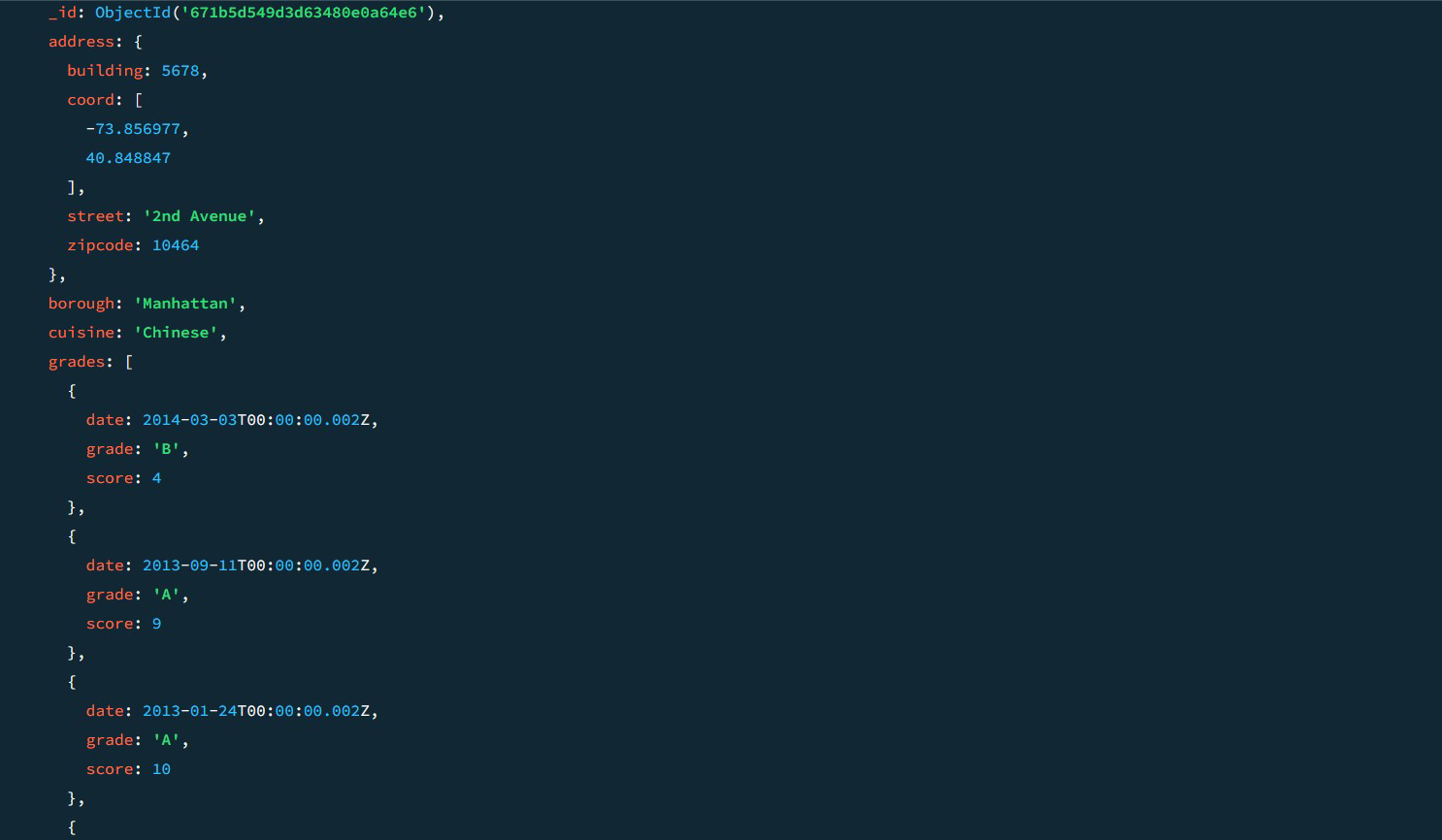
**{**

**"grades.score":{$lt:5},**

**borough: { $in: ["Manhattan", "Brooklyn"] },cuisine:{$ne:"American" }**

**}**

**);**



1. **Write a MongoDB query to find the restaurants that have at least one grade with ascoreoflessthan5andthatarelocatedintheboroughofManhattanorBrooklyn,andtheircuisine is notAmericanorChinese.**

**db.restaurants.find(**

**{**

**"grades.score":{$lt:5},**

**borough: { $in: ["Manhattan", "Brooklyn"] },cuisine:{$nin:["American","Chinese"]}**

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 andagrade withascore of6.**

**db.restaurants.find(**

**{**

**grades:{**

**$all:[**

**{$elemMatch:{score: 2}},**

**{$elemMatch:{score: 6}}**

**]**

**}**

**}**

**);**

**SAMPLEOUTPUT:-**

**{**

**\_id:ObjectId('671b92d339ec8a9bc8b6588b'),address:{**

**building: '1007',coord:[**

**-73.856077,**

**40.848447**

**],**

**street: 'Morris Park Ave',zipcode:'10462'**

**},**

**borough: 'Bronx',cuisine: 'Bakery',grades:[**

**{**

**date:2014-03-03T00:00:00.000Z,**

**grade: 'A',score:2**

**},**

**{**

**date:2013-09-11T00:00:00.000Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.000Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.000Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.000Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Morris Park Bake Shop',restaurant\_id:'30075445'**

**}**

**{**

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

**building: 1007,coord:[**

**-73.856077,**

**40.848447**

**],**

**street: 'Morris Park Ave',zipcode:10462**

**},**

**borough: 'Bronx',cuisine: 'Bakery',grades:[**

**{**

**date:2014-03-03T00:00:00.000Z,**

**grade: 'A',score:2**

**},**

**{**

**date:2013-09-11T00:00:00.000Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.000Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2011-11-23T00:00:00.000Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.000Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Morris Park Bake Shop',restaurant\_id:30075445**

**}**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 andagrade withascore of6andarelocatedintheborough ofManhattan.**

**db.restaurants.find(**

**{**

**borough: "Manhattan",grades:{**

**$all:[**

**{$elemMatch:{score: 2}},**

**{$elemMatch:{score: 6}}**

**]**

**}**

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 andagradewithascore of6andarelocatedintheborough ofManhattanorBrooklyn.**

**db.restaurants.find(**

**{**

**borough: { $in: ["Manhattan", "Brooklyn"] },grades:{**

**$all:[**

**{$elemMatch:{score: 2}},**

**{$elemMatch:{score: 6}}**

**]**

**}**

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 anda grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, andtheircuisine is notAmerican.**

**db.restaurants.find(**

**{**

**borough: { $in: ["Manhattan", "Brooklyn"] },grades:{**

**$all:[**

**{$elemMatch:{score: 2}},**

**{$elemMatch:{score: 6}}**

**]**

**},**

**cuisine:{$ne:"American"}**

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 anda grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, andtheircuisine is notAmericanorChinese.**

**db.restaurants.find(**

**{**

**borough: { $in: ["Manhattan", "Brooklyn"] },grades:{**

**$all:[**

**{$elemMatch:{score: 2}},**

**{$elemMatch:{score: 6}}**

**]**

**},**

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

**}**

**);**

1. **Write a MongoDB query to find the restaurants that have a grade with a score of 2 oragrade withascore of6.**

**db.restaurants.find(**

**{**

**$or:[**

**{"grades.score":2},**

**{"grades.score":6}**

**]**

**}**

**);**

**SAMPLEOUTPUT:-**

**{**

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

**building: 2233,coord:[**

**-73.858177,**

**40.849447**

**],**

**street: '5th Avenue',zipcode:10467**

**},**

**borough: 'Bronx',cuisine: 'American',grades:[**

**{**

**date:2014-03-03T00:00:00.005Z,**

**grade: 'A',score:10**

**},**

**{**

**date:2013-09-11T00:00:00.005Z,**

**grade: 'A',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.005Z,**

**grade: 'B',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.005Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2011-03-10T00:00:00.005Z,**

**grade: 'A',score:14**

**}**

**],**

**name: 'Burger Bistro',restaurant\_id:30075450**

**}**

**{**

**\_id:ObjectId('671b5dab56ec9972ca8f5daf'),address:{**

**building: 4455,coord:[**

**-73.858977,**

**40.849847**

**],**

**street: '7th Avenue',zipcode:10469**

**},**

**borough: 'Bronx',cuisine: 'Thai',grades:[**

**{**

**date:2014-03-03T00:00:00.007Z,**

**grade: 'A',score:9**

**},**

**{**

**date:2013-09-11T00:00:00.007Z,**

**grade: 'B',score:6**

**},**

**{**

**date:2013-01-24T00:00:00.007Z,**

**grade: 'A',score:12**

**},**

**{**

**date:2011-11-23T00:00:00.007Z,**

**grade: 'A',score:8**

**},**

**{**

**date:2011-03-10T00:00:00.007Z,**

**grade: 'B',score:14**

**}**

**],**

**name: 'Thai Delight',restaurant\_id:30075452**

**}**

**MOVIESCOLLECTION**

1. **Findallmovieswithfullinformationfromthe'movies'collectionthatreleasedintheyear**

**1893.**

**db.movies.find({year:1893});**

1. **Findallmovieswithfullinformationfromthe'movies'collectionthathavearuntimegreater**

**than120minutes.**

**db.movies.find({runtime:{$gt:120}});SAMPLEOUTPUT:-**

**{**

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

**plot:'AnastronautstrandedonMarsmustsurvivealone.',genres:[**

**'Sci-Fi','Drama'**

**],**

**runtime: 135,cast:[**

**'Matt Damon','JessicaChastain'**

**],**

**poster: 'https://m.media-amazon.com/images/poster4.jpg',title:'MarsAlone',**

**fullplot:'Anastronaut,leftaloneonMars,strugglestosurvivewith**

**limited resources while awaiting rescue.',languages:[**

**'English'**

**],**

**released:2015-10-02T00:00:00.000Z,**

**directors: ['RidleyScott'**

**],**

**rated: 'PG-13',awards: {wins:8,**

**nominations:6,**

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

**},**

**lastupdated:'2021-08-0917:22:30.000000000',**

**year: 2015,imdb: {rating:8,**

**votes:25650,**

**id:443**

**},**

**countries:['USA'**

**],**

**type: 'movie',tomatoes: {viewer:{**

**rating:4.5,**

**numReviews:2201,**

**meter:93**

**},**

**fresh: 18,critic: {rating:8.5,**

**numReviews:25,**

**meter:96**

**},**

**rotten:1,**

**lastUpdated:2021-07-19T21:20:55.000Z**

**}**

**}**

1. **Findallmovieswithfullinformationfromthe'movies'collectionthathave"Short" genre.**

**db.movies.find({ genres: "Short" });SAMPLEOUTPUT:-**

**{**

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

**plot:'Agroupofbanditsstageabrazentrainhold-up,onlytofinda**

**determined posse hot on their heels.',genres:[**

**'Short','Western'**

**],**

**runtime:11,cast:[**

**'A.C.Abadie',**

**"Gilbert M. 'Broncho Billy' Anderson",'GeorgeBarnes',**

**'JustusD.Barnes'**

**],**

**poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@.\_V1\_SY1000\_SX677\_AL\_.jpg',**

**title:'TheGreatTrainRobbery',**

**fullplot: "Among the earliest existing films in American cinema -notable as the first film that presented a narrative story to tell - itdepictsagroupofcowboyoutlawswhoholdupatrainandrobthe**

**passengers.TheyarethenpursuedbyaSheriff'sposse.Severalsceneshavecolorincluded-allhandtinted.",**

**languages: ['English'**

**],**

**released:1903-12-01T00:00:00.000Z,**

**directors:['EdwinS.Porter'**

**],**

**rated: 'TV-G',awards: {wins:1,**

**nominations:0,**

**text:'1win.'**

**},**

**lastupdated:'2015-08-1300:27:59.177000000',**

**year: 1903,imdb: {rating:7.4,**

**votes:9847,**

**id:439**

**},**

**countries:['USA'**

**],**

**type: 'movie',tomatoes: {viewer:{**

**rating:3.7,**

**numReviews:2559,**

**meter:75**

**},**

**fresh: 6,critic: {rating:7.6,**

**numReviews:6,**

**meter:100**

**},**

**rotten:0,**

**lastUpdated:2015-08-08T19:16:10.000Z**

**}**

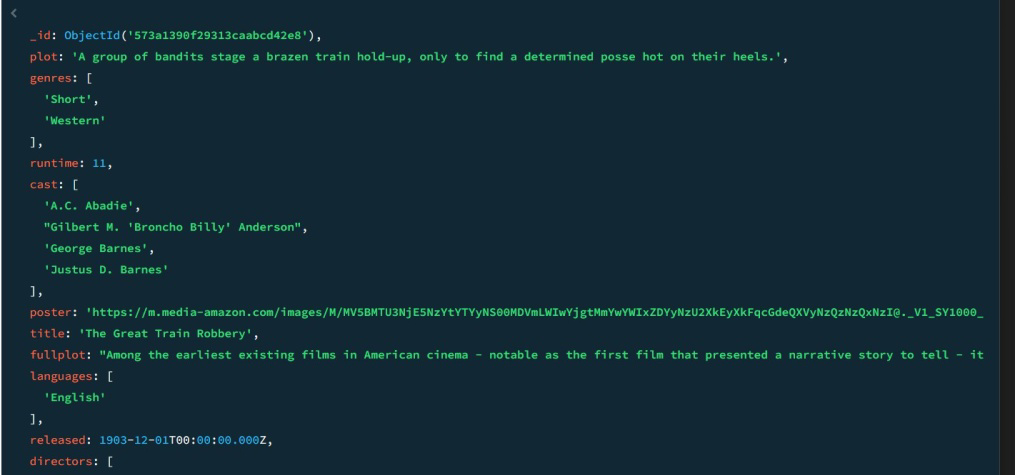
**}**

1. **Retrieve all movies from the 'movies' collection that were directedby "William K.L. Dickson"and include complete information for eachmovie.**

**db.movies.find({directors:"WilliamK.L.Dickson"});**

1. **Retrieveallmoviesfromthe'movies'collectionthatwerereleasedintheUSAandincludecompleteinformationforeachmovie.**

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



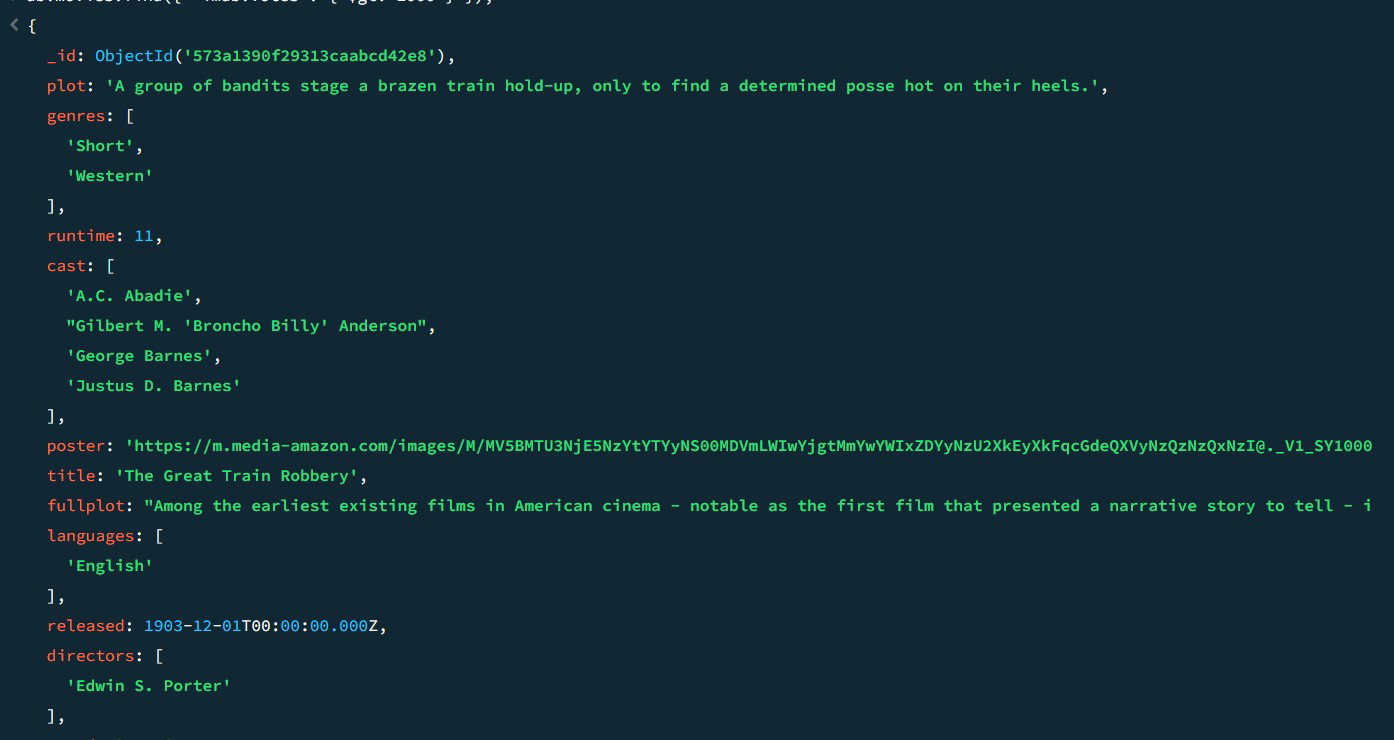
1. **Retrieveallmoviesfromthe'movies'collectionthathavecompleteinformationandarerated**

**as"UNRATED".**

**db.movies.find({rated:"UNRATED"});**

1. **Retrieveallmoviesfromthe'movies'collectionthathavecompleteinformationandhavereceivedmorethan1000votesonIMDb.**

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



1. **Retrieveallmoviesfromthe'movies'collectionthathavecompleteinformationandhaveanIMDb ratinghigherthan7.**

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



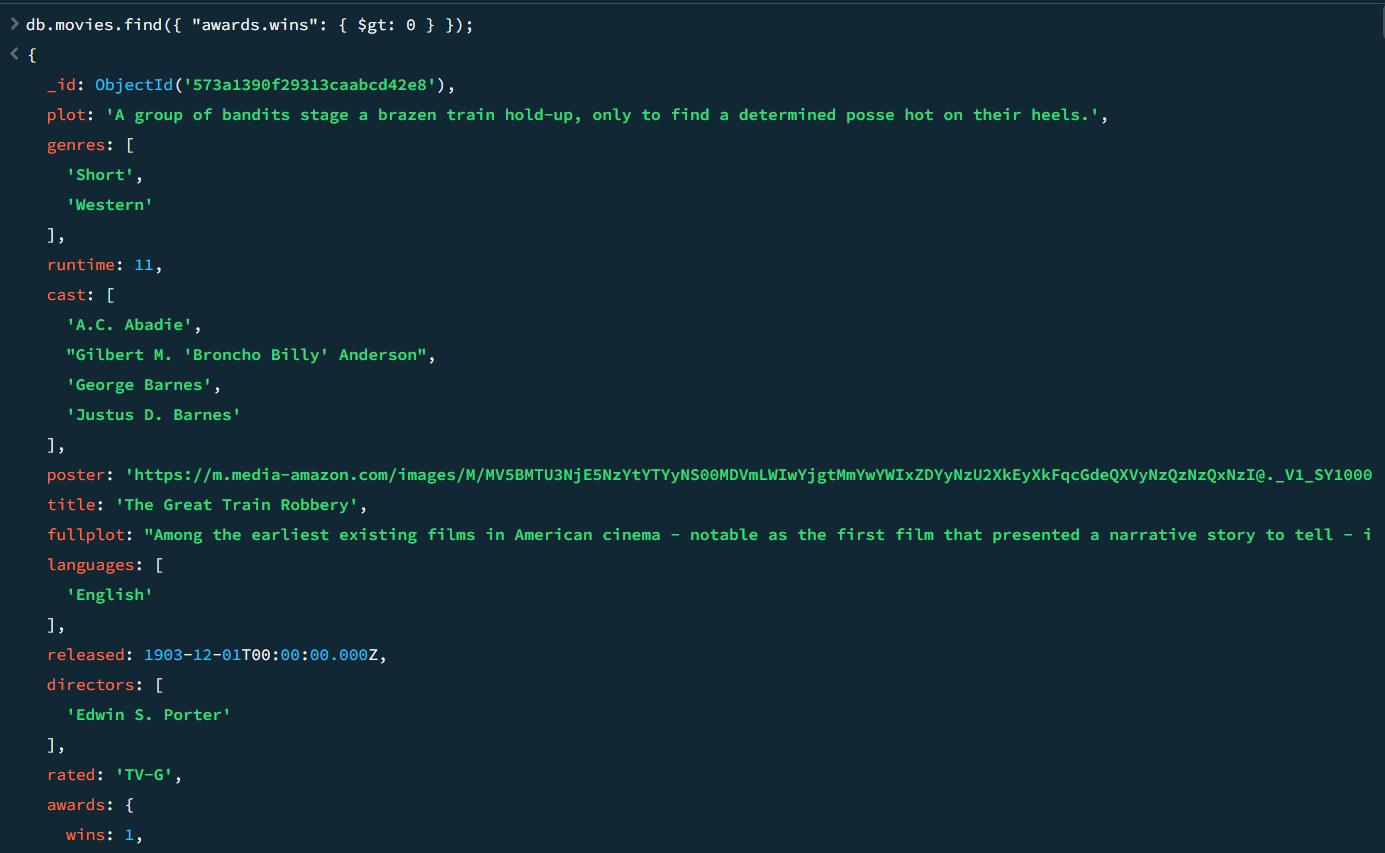
1. **Retrieve all movies from the 'movies' collection that havecompleteinformationandhaveaviewerratinghigherthan4onTomatoes.**

**db.movies.find({"tomatoes.viewer.rating":{$gt:4}});**



1. **Retrieveallmoviesfromthe'movies'collectionthathavereceivedanaward.**

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



1. **Findallmovieswithtitle,languages,released,directors,writers,awards, year, genres, runtime, cast, countries from the 'movies'collectioninMongoDBthathaveatleastonenomination.**

**db.movies.find(**

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

**{**

**title:1,**

**languages:1,**

**released:1,**

**directors:1,**

**writers:1,**

**awards:1,**

**year:1,**

**genres:1,**

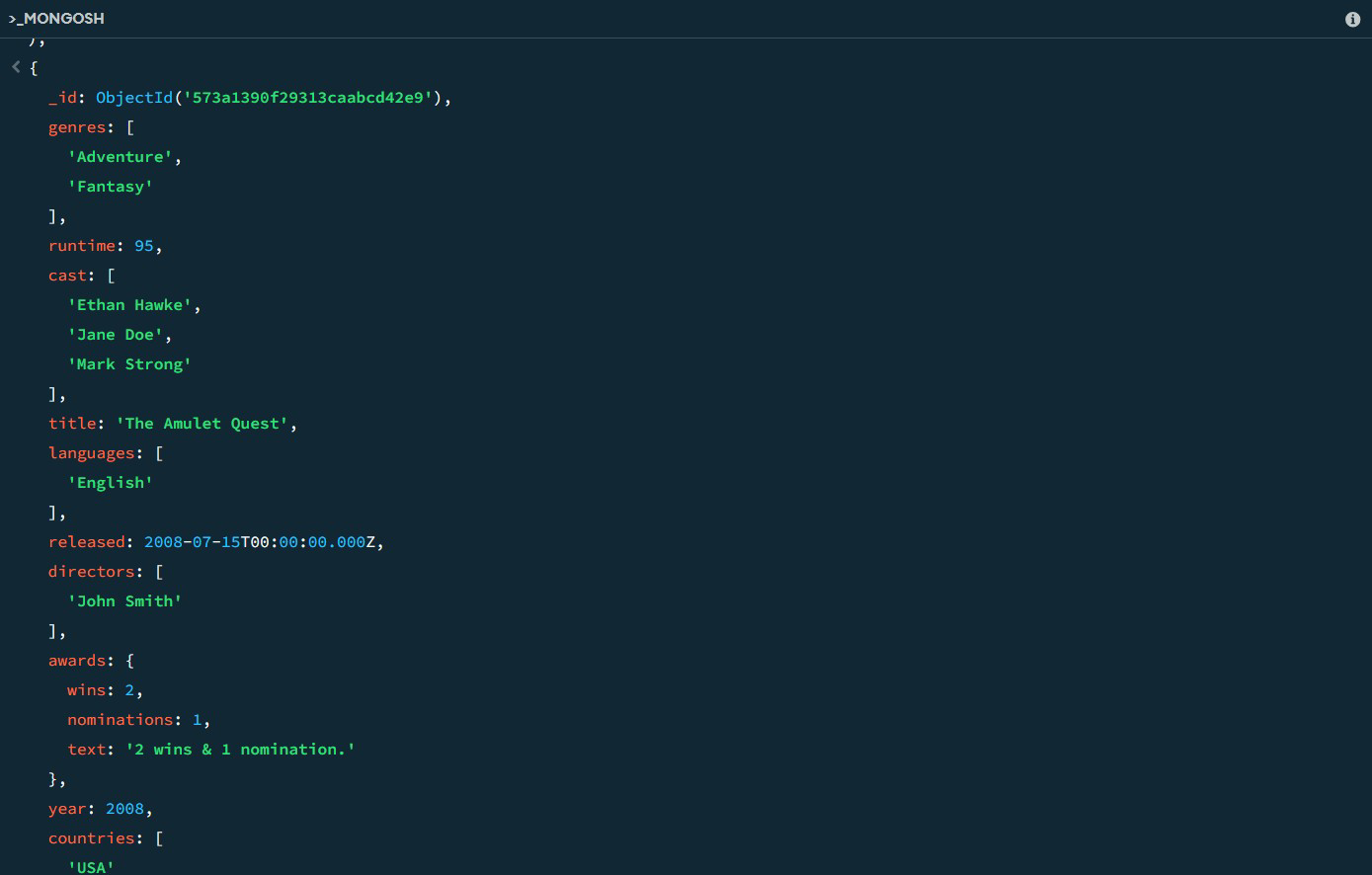
**runtime:1,**

**cast:1,**

**countries:1**

**}**

**);**



1. **Findallmovieswithtitle,languages,released,directors,writers,awards, year, genres, runtime, cast, countries from the 'movies'collectioninMongoDBwithcastincluding"CharlesKayser".**

**db.movies.find(**

**{cast:"CharlesKayser"},**

**{**

**title:1,**

**languages:1,**

**released:1,**

**directors:1,**

**writers:1,**

**awards:1,**

**year:1,**

**genres:1,**

**runtime:1,**

**cast:1,**

**countries:1**

**}**

**);**

1. **Retrieve all movies with title, languages, released, directors,writers, countries from the 'movies' collection in MongoDB thatreleasedonMay9,1893.**

**db.movies.find(**

**{released:ISODate("1893-05-09T00:00:00Z")},**

**{**

**title:1,**

**languages:1,**

**released:1,**

**directors:1,**

**writers:1,**

**countries:1**

**}**

**);**

1. **Retrieve all movies with title, languages, released, directors,writers,countriesfromthe'movies'collectioninMongoDBthathaveaword"scene"inthetitle.**

**db.movies.find(**

**{title:{$regex:/scene/i}},**

**{**

**title:1,**

**languages:1,**

**released:1,**

**directors:1,**

**writers:1,**

**countries:1**

**}**

**);**

|  |  |  |
| --- | --- | --- |
| **Ex.No.: 15** | | **OTHER DATABASE OBJECTS** |
| **Date:** | 27/09/2024 |

* 1. **Create a sequence to be used with the primary key column of the DEPT table. Thesequence should start at 200 and have a maximum value of 1000. Have your sequenceincrementbytennumbers.NamethesequenceDEPT\_ID\_SEQ.**

**CREATE SEQUENCE DEPT\_ID\_SEQSTARTWITH200**

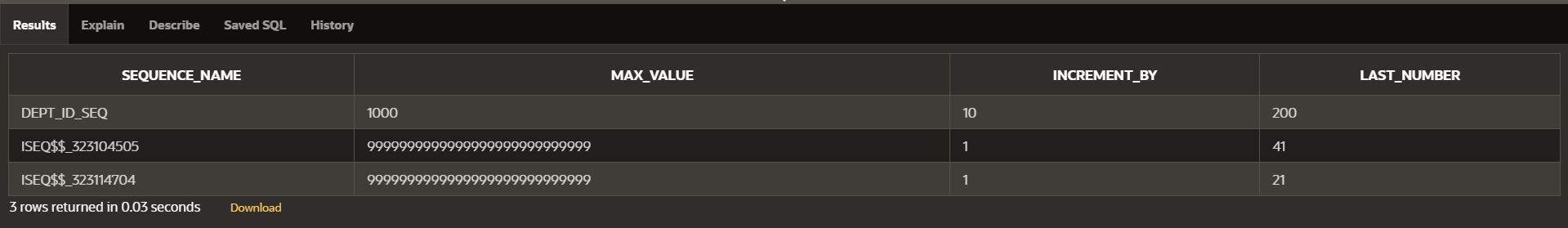
**INCREMENTBY10**

**MAXVALUE 1000NOCACHENOCYCLE;**

**2.Writeaqueryinascripttodisplaythefollowinginformationaboutyoursequences:sequencename, maximumvalue,increment size,andlastnumber**

**SELECT SEQUENCE\_NAME,MAX\_VALUE,INCREMENT\_BY,LAST\_NUMBER**

**FROMUSER\_SEQUENCES;**



**3WriteascripttoinserttworowsintotheDEPTtable.Nameyourscriptlab12\_3.sql.Besure to use the sequence that you created for the ID column. Add two departmentsnamed Education And Administration. Confirm your additions. Run the commands inyourscript.**

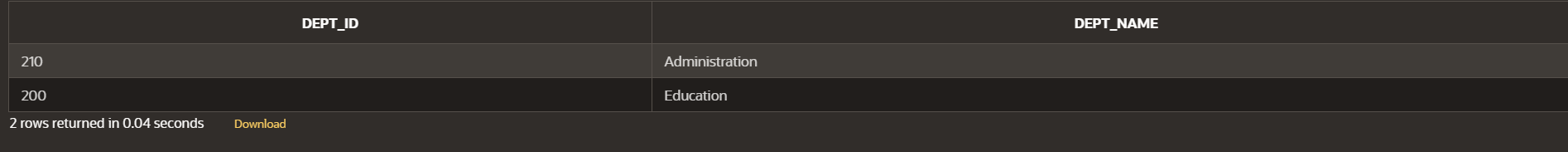
**INSERT INTO DEPT (DEPT\_ID, DEPT\_NAME)VALUES(DEPT\_ID\_SEQ.NEXTVAL,'Education');**

**INSERTINTODEPT(DEPT\_ID,DEPT\_NAME)**

**VALUES(DEPT\_ID\_SEQ.NEXTVAL,'Administration');**

**SELECT\*FROMDEPT**

**WHEREDEPT\_NAMEIN('Education','Administration');**



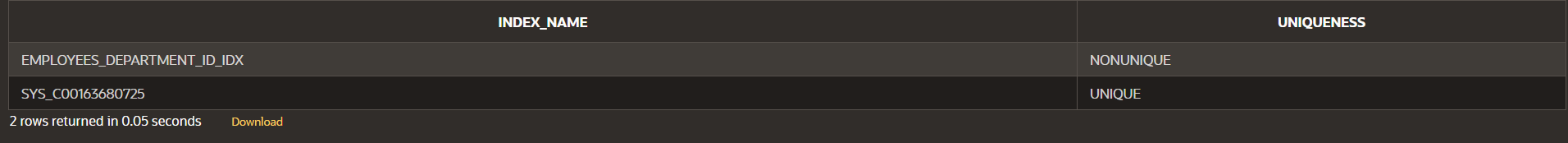
1. **Create a non unique index on the foreign key column (DEPARTMENT\_ID) in theEMPLOYEEStable.**

**CREATE INDEX employees\_department\_id\_idxONEMPLOYEES (DEPARTMENT\_ID);**

1. **DisplaytheindexesanduniquenessthatexistinthedatadictionaryfortheEMPtable.**

**SELECT INDEX\_NAME, UNIQUENESSFROMUSER\_INDEXES**

**WHERETABLE\_NAME='EMPLOYEES';**



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

1. **WhatprivilegeshouldauserbegiventologontotheOracleServer?Isthisasystemoranobjectprivilege?**

**TheprivilegeausershouldbegiventologontotheOracleServeristheCREATESESSIONprivilege.**

**Type of Privilege: This is a system privilege.GRANTCREATESESSIONTOusername;**

1. **Whatprivilegeshouldauserbegiventocreatetables?**

**theuserneedstheCREATETABLEprivilege.**

**TheCREATETABLEprivilegeallowstheusertocreatenewtablesintheirownschema.**

**GRANTCREATETABLETOusername;**

1. **Ifyoucreateatable,whocanpassalongprivilegestootherusersonyourtable?**

**When you create a table, only you as the table owner (or a user with the ADMIN OPTIONor GRANT ANY PRIVILEGE system privilege) can grant privileges on your table to otherusers.**

**GRANTSELECTONyour\_tableTOother\_user;**

1. **You are the DBA. You are creating many users who require the same systemprivileges.Whatshould youuse tomakeyour jobeasier?**

**AsaDBA,tosimplifytheprocessofgrantingthesamesystemprivilegestomultipleusers,you should use roles.**

**CREATEROLEmy\_role;**

**GRANT CREATE SESSION TO my\_role;GRANTCREATETABLE TOmy\_role;**

**GRANTmy\_roleTOuser1;GRANTmy\_roleTOuser2;**

1. **Whatcommanddoyouusetochangeyourpassword?**

**ALTERUSERusernameIDENTIFIEDBYnew\_password;**

1. **Grant another user access to your DEPARTMENTS table. Have the user grant youqueryAccess tohis orher DEPARTMENTS table.**

**Grant Accessto Your DEPARTMENTSTable**

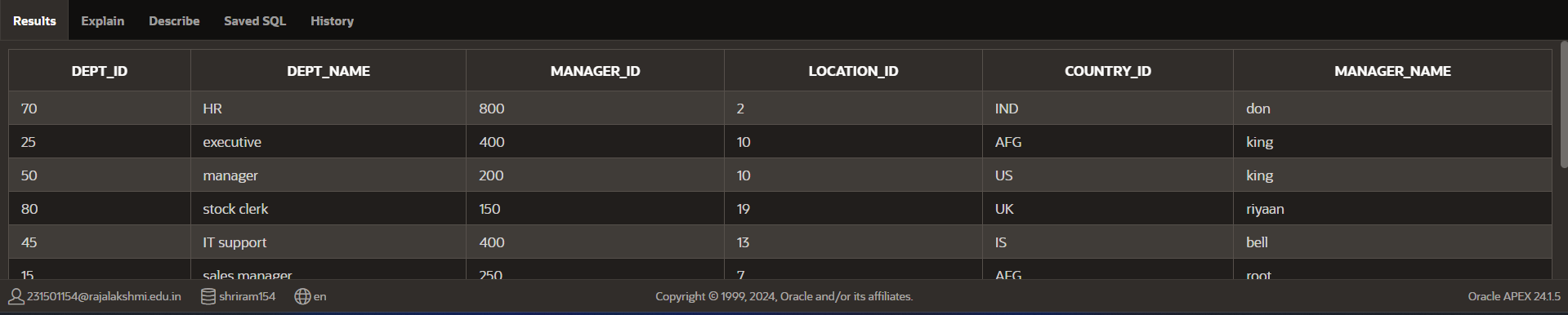
**GRANTSELECTONyour\_username.DEPARTMENTSTOother\_user;**

**Grant Query Access toOtherUser's DEPARTMENTSTable**

**GRANTSELECTONother\_user.DEPARTMENTSTOyour\_username;**

1. **QueryalltherowsinyourDEPARTMENTStable.**

**SELECT\*FROMDEPARTMENT;**



1. **Add a new row to your DEPARTMENTS table. Team 1 should add Education asdepartment number 500. Team 2 should add Human Resources department number 510.Querytheotherteam‘stable.**

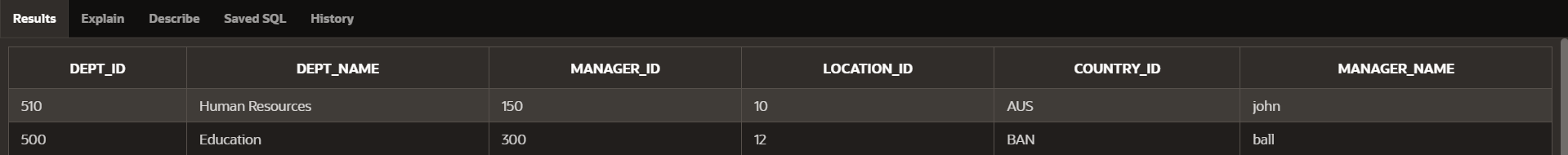
**INSERTINTODEPARTMENT(dept\_id,**

**DEPT\_NAME,manager\_id,location\_id,country\_id,manager\_name)VALUES(500,'Education',300,12,'BAN','ball');**

**INSERTINTODEPARTMENT(dept\_id,**

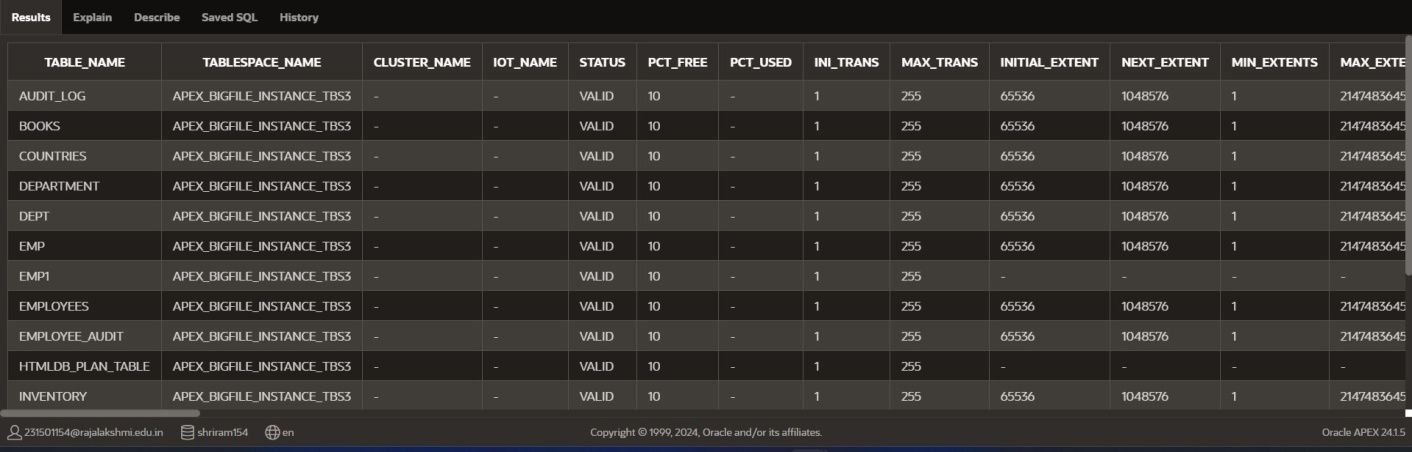
**DEPT\_NAME,manager\_id,location\_id,country\_id,manager\_name)VALUES(510,'HumanResources',150,10,'AUS','john');**

**SELECT\*FROMDEPARTMENT;**



1. **Query the USER\_TABLES data dictionary to see information about the tables that youown.**

**SELECT\*FROMUSER\_TABLES;**



1. **RevoketheSELECTprivilegeonyourtablefromtheotherteam.**

**REVOKESELECTONteam1\_user.DEPARTMENTSFROMother\_user;**

1. **Remove the row you inserted into the DEPARTMENTS table in step 8 and save thechanges.**

**DELETE FROM DEPARTMENTWHEREDEPT\_IDIN(500,510);**