KATARINA CEHOVSKI

ASSIGNMENT A

**Request 1**

List the last name, first name and employee number of all programmers who were hired on or before 21 May 1991 sorted in ascending order of last name.

**Expected**

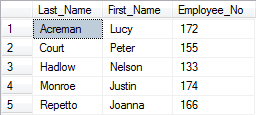
3 columns   
Last\_Name, First\_Name, Employee\_No  
Cehovski, Katarina, 123  
…

**Query**

SELECT Employees.Last\_Name, Employees.First\_Name, Employees.Employee\_No

FROM Employees JOIN Jobs ON Jobs.Job\_ID= Employees.Job\_ID WHERE

Jobs.Job\_Title = 'Programmer' AND Hire\_Date <= '1991-05-21' ORDER BY Last\_Name ASC

****

**Request 2**

List the department number, last name and salary of all employees who were hired between 16/09/87 and 12/05/96 sorted in ascending order of last name within department number.

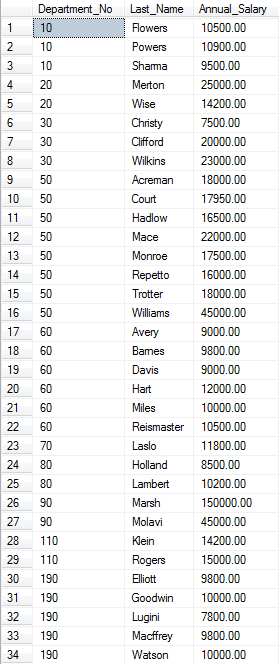
**Expected**

3 columns   
Department\_No, Last\_Name , Annual\_Salary  
120, Cehovski, 14000  
…

**Query**

SELECT Department\_No, Last\_Name, Annual\_Salary FROM Employees WHERE

Hire\_Date BETWEEN '1987-09-16' AND '1996-05-12' ORDER BY Department\_No, Last\_Name ASC

****

**Request 3**

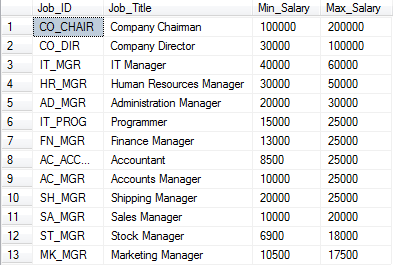
List all the data for all jobs where the maximum salary is greater than 15000 sorted in descending order of the maximum salary.

**Expected**

4 columns  
Job\_ID, Job\_Title, Min\_Salary, Max\_Salary  
IT\_SOF\_DEV, Software Developer, 15000, 25000   
…

**Query**

SELECT \* FROM Jobs WHERE Max\_Salary> 15000 ORDER BY Max\_Salary DESC

****

**Request 4**

List the last name, first name, job id and commission of employees who earn commission sorted in ascending order of first name within last name.

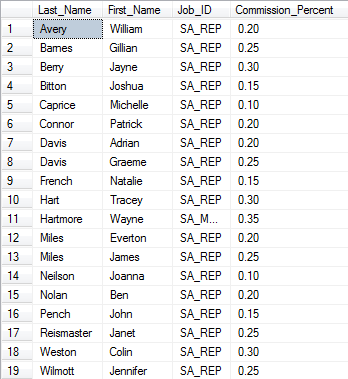
**Expected**

4 columns  
Last\_Name, First\_Name, Job\_ID, Commission\_Percent  
Cehovski, Katarina, IT\_SOF\_DEV, 0.15  
…

**Query**

SELECT Last\_Name, First\_Name, Job\_ID, Commission\_Percent FROM Employees WHERE

Commission\_Percent IS NOT NULL ORDER BY Last\_Name, First\_Name ASC



**Request 5**

Which jobs are found in the IT and Sales departments?

**Expected**

1 column   
Job\_Title   
Programmer  
…

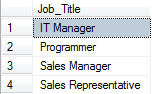
**Query**

SELECT DISTINCT Job\_Title FROM Employees

JOIN Jobs ON Employees.Job\_ID = Jobs.Job\_ID

JOIN Departments ON Departments.Department\_No = Employees.Department\_No

WHERE Department\_Name ='IT' OR Department\_Name ='Sales'



**Request 6**

List the last name of all employees in departments 50 and 90 together with their monthly salaries (rounded to 2 decimal places), sorted in ascending order of last name.

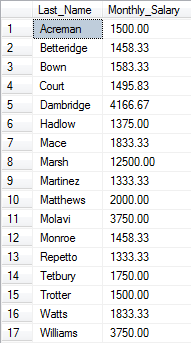
**Expected**

2 columns  
Last\_Name, Montly\_Salary  
Cehovski, 1400.43  
…

**Query**

SELECT Last\_Name, CAST(ROUND(Annual\_Salary/12, 2) AS DECIMAL(8,2)) AS Monthly\_Salary FROM Employees WHERE

Department\_No = 50 or Department\_No=90 ORDER BY Last\_Name ASC



**Request 7**

Show the total salaries figure for one month displayed with no decimal places.

**Expected**

1 column with number  
Monthly\_Salary  
10201

**Query**

SELECT CAST(SUM(Annual\_Salary/12) AS INT) AS Monthly\_Salary FROM Employees



**Request 8**

Show the total number of employees.

**Expected**

1 column with number  
Number\_of\_Employees  
64

**Query**

SELECT COUNT(Employee\_No) AS Number\_of\_Employees FROM Employees



**Request 9**

List the department number, department name and the number of employees for each department that has more than 2 employees grouping by department number and department name.

**Expected**

3 columns   
Department\_No, Department\_Name, Number\_of\_Employees  
120, IT, 15  
…

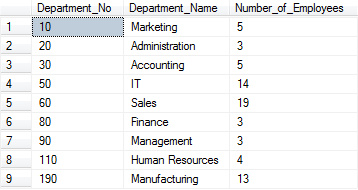
**Query**

SELECT Departments.Department\_No, Departments.Department\_Name, COUNT(\*) AS Number\_of\_Employees

FROM Departments INNER JOIN Employees ON Departments.Department\_No= Employees.Department\_No

GROUP BY Departments.Department\_No, Departments.Department\_Name

HAVING COUNT(\*) >2



**Request 10**

List the department number, department name and the number of employees for the department that has the highest number of employees using appropriate grouping.

**Expected**

3 columns  
Department\_No, Department\_Name, Number\_Employee  
150, Finance, 25

**Query**

SELECT TOP 1 Departments.Department\_No, Departments.Department\_Name,

COUNT(\*) AS Number\_Employee

FROM Departments INNER JOIN Employees ON Departments.Department\_No = Employees.Department\_No

GROUP BY Departments.Department\_Name, Departments.Department\_No

ORDER BY Number\_Employee DESC



**Request 11**

List the department number and name for all departments where no programmers work.

**Expected**

2 columns  
Department\_No, Department\_Name  
50, Sales  
…

**Query**

SELECT Departments.Department\_No, Departments.Department\_Name

FROM Departments

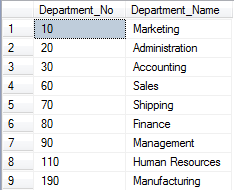
WHERE Departments.Department\_No not in(

SELECT DISTINCT Departments.Department\_No FROM Employees

JOIN Jobs ON Employees.Job\_ID = Jobs.Job\_ID

JOIN Departments ON Employees.Department\_No = Departments.Department\_No

AND Jobs.Job\_Title= 'Programmer')



**Request 12**

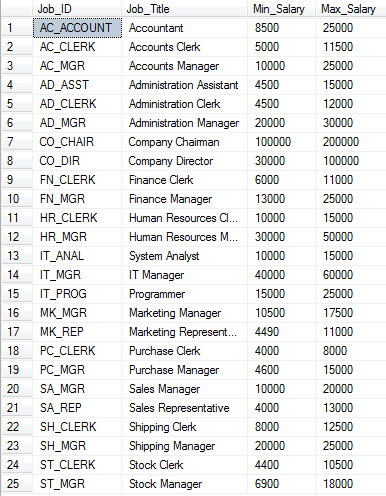
Add the following new job IT\_ANAL, System Analyst, 10000, 15000

**Expected**

New row will be added in table Jobs:  
Job\_ID, Job\_Title, Min\_Salary, Max\_Salary  
IT\_ANAL, System Analyst, 10000, 15000

**Query**

INSERT INTO Jobs VALUES ('IT\_ANAL', 'System Analyst', 10000, 15000)



**Request 13**

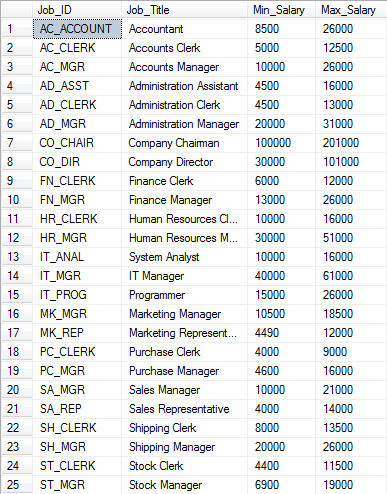
Update all the maximum salaries for jobs with an increase of 1000.

**Expected**

4 columns  
column Max\_Salary -> Before: 15000, After: 16000

**Query**

UPDATE Jobs SET Max\_Salary += 1000



**Request 14**

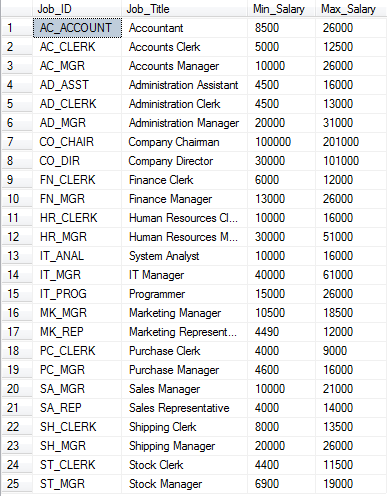
List all the data for jobs sorted in ascending order of job id.

**Expected**

4 columns  
Job\_ID, Job\_Title, Min\_Salary, Max\_Salary  
AC\_ACCOUNT, Accountant, 8500, 26000

**Query**

SELECT \* FROM Jobs ORDER BY Job\_ID ASC



**Request 15**

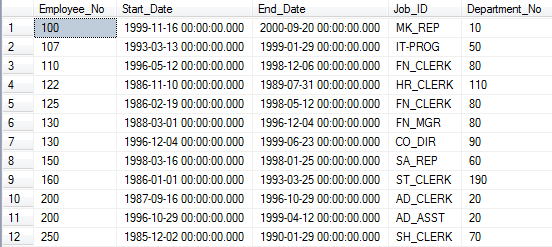
1. The job history for employee number 102 is no longer required. Delete this record.

**Expected**

Employee with number 102 will no longer exist in table Job\_History.

**Query**

DELETE FROM Job\_History WHERE Employee\_No= 102



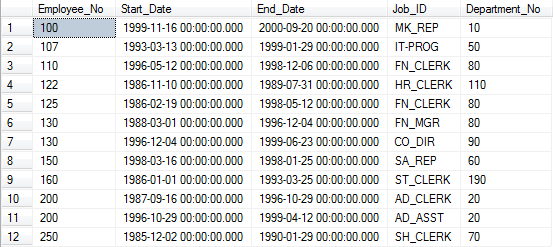
1. List all the data for job history sorted in ascending order of employee number.

**Expected**

5 columns  
Employee\_No, Start\_Date, End\_Date, Job\_ID, Department\_No  
10, 12/05/1998, 20/08/2011, AD\_ASST, 20

**Query**

SELECT \* FROM Job\_History ORDER BY Employee\_No ASC



**Request 16**

Produce a list of employees showing percentage raises, employee numbers and old and new salaries. Employees in departments 20 and 10 are given a 5% rise, employees in departments 50, 80, 90 and 110 are given a 10% rise and employees in other departments are not given a rise.

**Expected**

4 columns  
Employee\_No, Old\_Salary, Raise, New\_Salary  
123, 17000.00, 0.05, 17850.0000

**Query**

select Employees.Employee\_No, Employees.Annual\_Salary AS Old\_Salary,

(

CASE

WHEN Employees.Department\_No in (10,20) THEN 0.05

WHEN Employees.Department\_No in (50,80,90,110) THEN 0.1

ELSE 0

END

) as Raise,

(Employees.Annual\_Salary \* (1+

(

CASE

WHEN Employees.Department\_No in (10,20) THEN 0.05

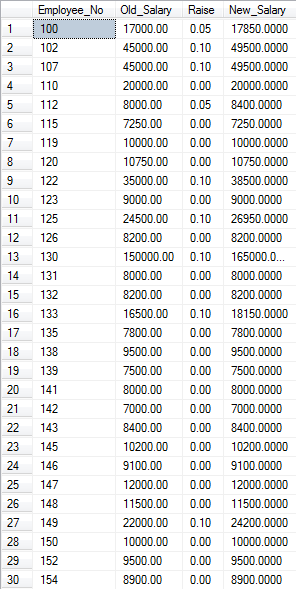
WHEN Employees.Department\_No in (50,80,90,110) THEN 0.1

ELSE 0

END

))) as New\_Salary

from Employees

****

**Request 17**

Create a new view for manager’s details only using all the fields from the employee table. Apply a CHECK constraint.

**Expected**

New view table with 11 columns. Names of the columns will be the same like in table Employees, and it will contain details just for employees who are managers.

**Query**

CREATE VIEW Managers\_Details AS

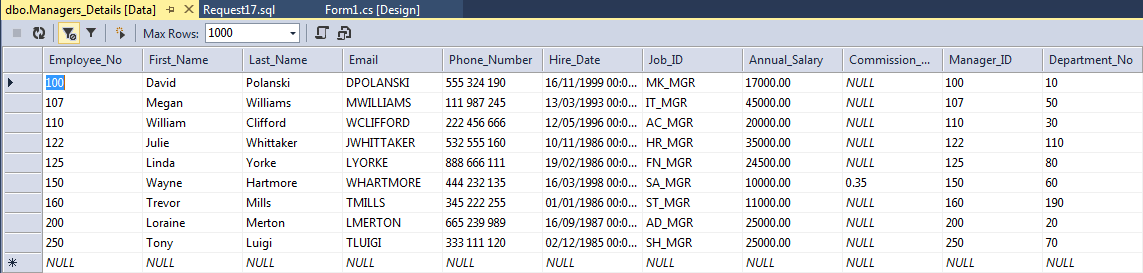
(

SELECT \* FROM Employees

WHERE Job\_ID LIKE '%MGR'

)

WITH CHECK OPTION

****

**Request 18**

Show all the fields and all the managers using the view for managers sorted in ascending order of employee number.

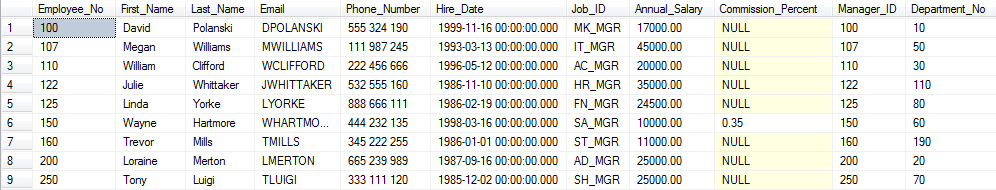
**Expected**

11 columns  
Employee\_No, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_Date, Annual\_Salary, Manager\_ID, Department\_No  
123, Katarina, Cehovski, KCEHO, 452 542 546, 12/12/2012, IT\_SOFT\_DEV, 15000.00, 125, 50

**Query**

SELECT \* FROM Managers\_Details

ORDER BY Employee\_No ASC

****

**Request 19**

Grant the authority to all other users to access the view for managers for SELECT statements only.

**Expected**

All users will be able to use only SELECT statement .

**Query**

GRANT SELECT ON Managers\_Details TO PUBLIC

****

**Request 20**

Create an index named LOC\_POSTAL\_CODEon the Postal Code in the locations table. Provide a printout showing that the index has been created.

**Expected**

New index will be added.

**Query**

CREATE INDEX LOC\_POSTAL\_CODE ON Locations (Postal\_Code)

