/\*

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.

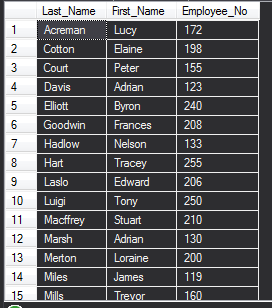
\*/

select Last\_Name, First\_Name, Employee\_No from Employees where Hire\_Date <= '1991/05/21' order by Last\_Name asc;

Expected:

|  |  |  |
| --- | --- | --- |
| Last\_Name | First\_Name | Employee\_No |
| John | Smith | 101 |

Result:



/\*

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.

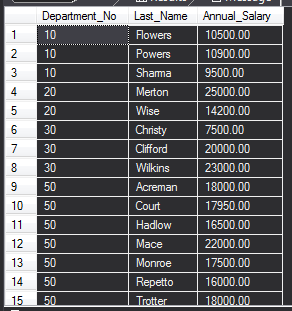
\*/

select Employees.Department\_No, Employees.Last\_Name, Employees.Annual\_Salary from Employees where Hire\_Date between '1987/09/16' AND '1996/05/12' order by Department\_No, Last\_Name asc;

Expected:

|  |  |  |
| --- | --- | --- |
| Department\_No | Last\_Name | Annual\_Salary |
| 10 | Smith | 10000.00 |

Result:



/\*

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.

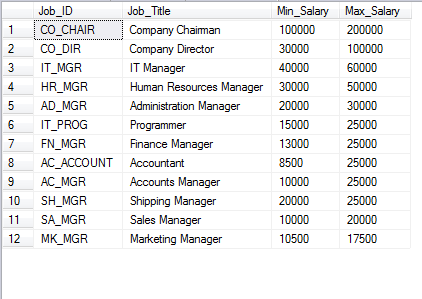
\*/

Select \* from Jobs where Max\_Salary > 15000 ORDER BY Max\_Salary DESC;

Expected:

|  |  |  |  |
| --- | --- | --- | --- |
| Job\_ID | Job\_Title | Min\_Salary |  |
| AC\_ACCOUNT | Accountant | 10000.00 |  |

Result:



/\*

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.

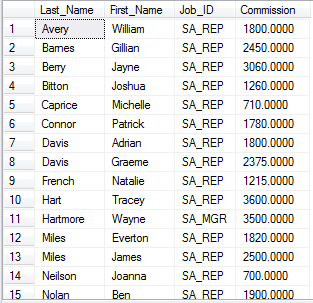
\*/

SELECT Employees.Last\_Name, Employees.First\_Name, Employees.Job\_ID, (Employees.Annual\_Salary \* Employees.Commission\_Percent) as Commission from Employees where Commission\_Percent IS NOT NULL order by Last\_Name, First\_Name;

Expected:

|  |  |  |  |
| --- | --- | --- | --- |
| Last\_Name | First\_Name | Job\_ID | Commission |
| Smith | John | AC\_ACCOUNT | 10000.00 |

Result:



/\*

Request 5

Which jobs are found in the IT and Sales departments?

\*/

Select distinct Jobs.Job\_Title from Jobs

Join Employees on Employees.Job\_ID = Jobs.Job\_ID

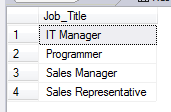
Join Departments on Departments.Department\_No = Employees.Department\_No

where Departments.Department\_Name = 'IT' OR Departments.Department\_Name = 'Sales';

Expected:

|  |
| --- |
| Job\_Title |
| AC\_ACCOUNT |

Result:



/\*

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.

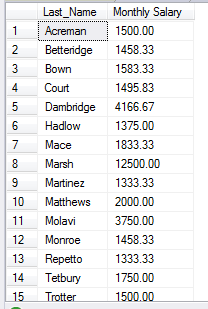
\*/

Select Employees.Last\_Name, cast(round((Employees.Annual\_Salary / 12), 2) as decimal(10, 2)) as 'Monthly Salary' from Employees where Employees.Department\_No = 50 OR Employees.Department\_No = 90 order by Last\_Name asc;

Expected:

|  |  |
| --- | --- |
| Last\_Name | Monthly\_Salary |
| Smith | 1000.00 |

Result:



/\*

Request 7

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

\*/

Select cast(sum(Employees.Annual\_Salary / 12) as int) as 'Total Monthly Salary' from Employees;

Expected:

|  |
| --- |
| Total Monthly Salary |
| 100000 |

Result:



/\*

Request 8

Show the total number of employees.

\*/

select count(\*) as 'Total Employees' from Employees;

Expected:

|  |
| --- |
| Total Employees |
| 100 |

Result:



/\*

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.

\*/

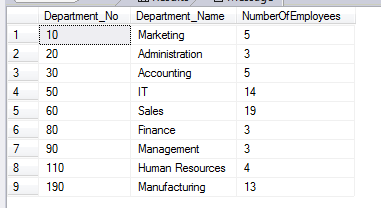
SELECT Employees.Department\_No, Departments.Department\_Name, count(\*) as NumberOfEmployees from Employees

JOIN Departments on Departments.Department\_No = Employees.Department\_No group by Employees.Department\_No, Department\_Name having count(\*) > 2;

Expected:

|  |  |  |
| --- | --- | --- |
| Department\_No | Department\_Name | NumberOfEmployees |
| 1 | Management | 1 |

Result:



/\*

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.

\*/

select top 1 Employees.Department\_No, Departments.Department\_Name, count(\*) as NumberOfEmployees from Employees

Join Departments on Departments.Department\_No = Employees.Department\_No group by Employees.Department\_No, Departments.Department\_Name order by count(\*) desc;

Expected:

|  |  |  |
| --- | --- | --- |
| Department\_No | Department\_Name | NumberOfEmployees |
| 1 | Management | 1 |

Result:



/\*

Request 11

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

\*/

Select distinct Departments.Department\_No, Departments.Department\_Name from Departments

Join Employees on Departments.Department\_No = Employees.Department\_No

Join Jobs on Jobs.Job\_ID = Employees.Job\_ID

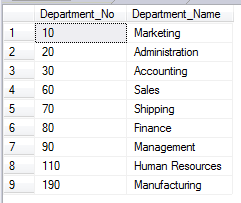
where Departments.Department\_No not in (Select distinct Department\_No from Employees where Employees.Job\_ID like '%PROG%');

Expect department 50 not to be listed

Expected:

|  |  |
| --- | --- |
| Department\_No | Department\_Name |
| 1 | Management |

Result:



/\*

Request 12

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

\*/

insert into jobs(Job\_ID, Job\_Title, Min\_Salary, Max\_Salary) values('IT\_ANAL', 'System Analyst', 10000, 15000);

Expect IT\_ANAL to appear in Jobs Table

Result:





/\*

Request 13 Update all the maximum salaries for jobs with an increase of 1000.

\*/

update jobs set Max\_Salary += 1000;

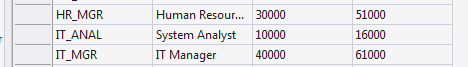
Expect Max\_Salary to increase by 1000

Result:

Before:



After:



/\*

Request 14 List all the data for jobs sorted in ascending order of job id.

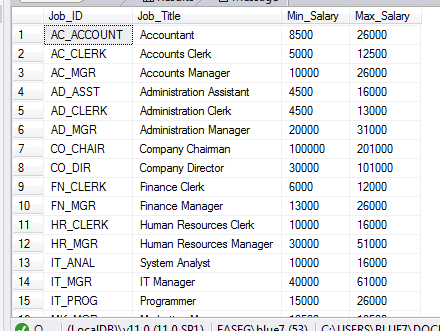
\*/

select \* from jobs order by Job\_ID Asc;

Expected:

|  |  |  |  |
| --- | --- | --- | --- |
| Job\_ID | Job\_Title | Min\_Salary | Max\_Salary |
| AC\_ACCOUNT | Accountant | 10000 | 15000 |

Result:



/\*

Request 15

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

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

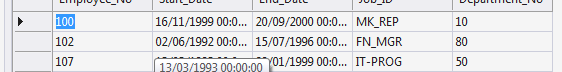
\*/

delete from Job\_History where Employee\_No=102;

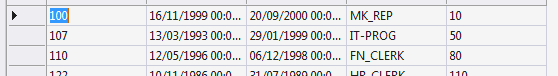
Expect record for Employee\_No to disappear

Result:

Before:



After:

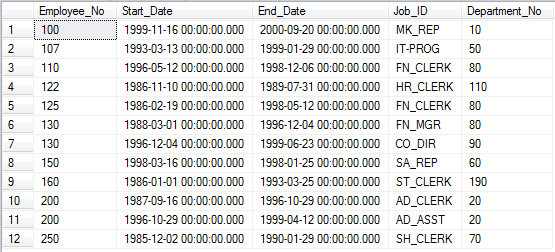


Select \* from Job\_History order by Employee\_No;

Expected:

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_No | Start\_Date | End\_Date | Job\_ID |
| AC\_ACCOUNT | Accountant | 10000 | 15000 |

Result:



/\*

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.

\*/

select

CASE

WHEN Employees.Department\_No=20 OR Employees.Department\_No=10

THEN 5

WHEN Employees.Department\_No=50 OR Employees.Department\_No=80

OR Employees.Department\_No=90 OR Employees.Department\_No=110 THEN 10

ELSE 0

END as 'Percent\_Increase(%)', Employees.Employee\_No, Employees.Annual\_Salary,

CAST(((

CASE

WHEN Employees.Department\_No=20 OR Employees.Department\_No=10 THEN 5 / 100.0

WHEN Employees.Department\_No=50 OR Employees.Department\_No=80

OR Employees.Department\_No=90 OR Employees.Department\_No=110 THEN 10 / 100.0

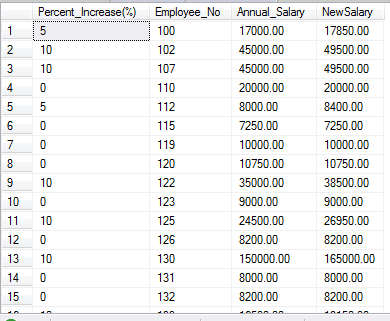
ELSE 0

END) \* Annual\_Salary) + Annual\_Salary as decimal(10, 2)) as NewSalary from Employees;

Expected:

|  |  |  |  |
| --- | --- | --- | --- |
| Percent\_Increase(%) | Employee\_No | Annual\_Salary | NewSalary |
| 5 | 1 | 10000 | 10500 |

Result:



/\*

Request 17

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

\*/

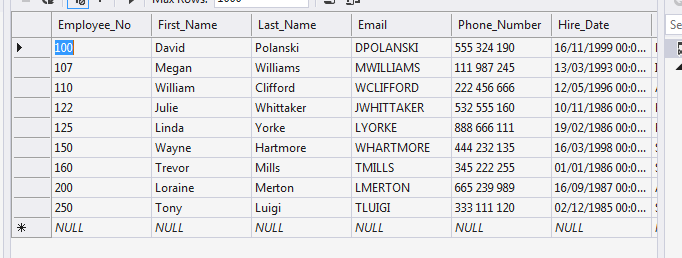
CREATE VIEW Managers

as SELECT \* FROM Employees where Employees.Job\_ID like '%MGR%' WITH Check Option;

Expect view called managers to be created.

Result:





/\*

Request 18

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

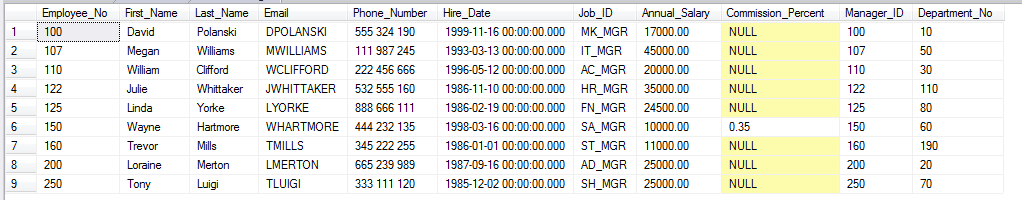
\*/

SELECT \* FROM Managers;

Expected:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Employee\_No | First\_Name | Last\_Name | Email | Phone\_Number | Hire\_Date | Job\_ID | Annual\_Salary | Commission\_Percent | Manager\_ID | Department\_No |
| 1 | John | Smith | JSMITH | 000 123 456 | 1999-01-01 00.00.00.000 | AC\_ACCOUNT | 10000.00 | NULL | 1 | 1 |

Result:



/\*

Request 19

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

\*/

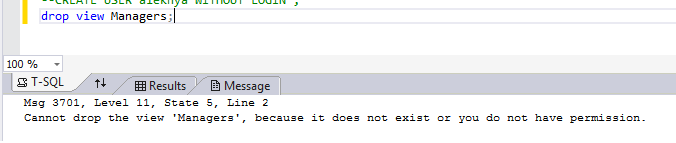
GRANT SELECT ON Managers TO PUBLIC;

Expect only select to work for queries run not by admin not drop etc

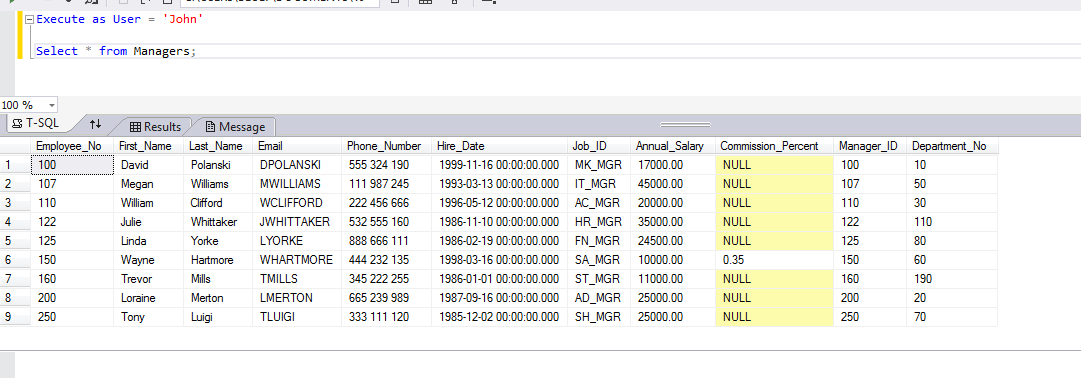


Result:

Can’t Drop:



Can Select:



/\*

Request 20

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

\*/

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

Expect an index to appear in the table data view.

Result: