# Lab: Databases and SQL Language

You can check your solutions here: <https://judge.softuni.org/Contests/3136/Additional-Exercises>.

## Display All Information about the Departments

Write a SQL query to find **all available information about the Departments**.

### Example

|  |  |  |
| --- | --- | --- |
| **DepartmentID** | **Name** | **ManagerID** |
| 1 | Engineering | 12 |
| 2 | Tool Design | 4 |
| 3 | Sales | 273 |
| … | … | … |

### Hints

Graphical user interface, table

Description automatically generated

## Display All Department Names

Write SQL query to find **all Department names**.

### Example

|  |
| --- |
| **Name** |
| Engineering |
| Tool Design |
| Sales |
| … |

### Hints

Graphical user interface, application

Description automatically generated

## Salary of Each Employee

Write SQL query to find the **first name**, **last name** and **salary** of each employee.

### Example

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Salary** |
| Guy | Gilbert | 12500.00 |
| Kevin | Brown | 13500.00 |
| Roberto | Tamburello | 43300.00 |
| … | … | … |

## All Different Employee’s Salaries

Write a SQL query to find **all different employee’s salaries**. Show only the salaries.

### Example

|  |
| --- |
| **Salary** |
| 9000.00 |
| 9300.00 |
| 9500.00 |
| … |

## Names of All Employees by Salary in Range

Write a SQL query to find the **first name**, **last name** and **job title** of all employees whose **salary is in the** **range [20000, 30000].**

### Example

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **JobTitle** |
| Rob | Walters | Senior Tool Designer |
| Thierry | D'Hers | Tool Designer |
| JoLynn | Dobney | Production Supervisor |
| … | … | … |

## All Employees Without Manager

Write a SQL query to find **first and last names** about those employees that **does not have a manager**.

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Ken | Sanchez |
| Svetlin | Nakov |
| … | … |

## All Employees with Salary More Than 50000

Write a SQL query to find **first name**, **last name** and **salary** of those employees who has salary more than 50000. Order them in decreasing order by salary.

### Example

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Salary** |
| Ken | Sanchez | 125500.00 |
| James | Hamilton | 84100.00 |
| … | … | … |

## 5 Best Paid Employees.

Write SQL query to find **first and last names** about **5 best paid Employees** ordered **descending by their salary.**

### Example

|  |  |
| --- | --- |
| **FirstName** | **LastName** |
| Ken | Sanchez |
| James | Hamilton |
| … | … |

## Last 7 Hired Employees

Write a SQL query to find **last 7 hired employees**. Select **their first, last name and their hire date**.

### Example

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **HireDate** |
| Rachel | Valdez | 2005-07-01 00:00:00 |
| Lynn | Tsoflias | 2005-07-01 00:00:00 |
| Syed | Abbas | 2005-04-15 00:00:00 |
| … | … | … |

## Increase Salaries

Write a SQL query to increase salaries of all employees that are in the **Engineering**, **Tool Design**, **Marketing** or **Information Services** department by **12%**. Then **select Salaries column** from the **Employees** table.

### Example

|  |
| --- |
| **Salary** |
| 12500.00 |
| 15120.00 |
| 48496.00 |
| 33376.00 |
| … |

## Employee Address

Write a query that selects:

* **EmployeeId**
* **JobTitle**
* **AddressId**
* **AddressText**

Return the **first 5** rows **sorted** by **AddressId** in **ascending** order.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeId** | **JobTitle** | **AddressId** | **AddressText** |
| 142 | Production Technician | 1 | 108 Lakeside Court |
| 30 | Human Resources Manager | 2 | 1341 Prospect St |
| … | … | … | … |

## Addresses with Towns

Write a query that selects:

* **FirstName**
* **LastName**
* **Town**
* **AddressText**

**Sorted** by **FirstName** in **ascending** order then by **LastName**. Select **first 50** employees.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstName** | **LastName** | **Town** | **AddressText** |
| A.Scott | Wright | Newport Hills | 1400 Gate Drive |
| Alan | Brewer | Kenmore | 8192 Seagull Court |
| … | … | … | … |

## Sales Employee

Write a query that selects:

* **EmployeeID**
* **FirstName**
* **LastName**
* **DepartmentName**

**Sorted** by **EmployeeID** in **ascending** order. Select only **employees** from “**Sales**” department.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **DepartmentName** |
| 268 | Stephen | Jiang | Sales |
| 273 | Brian | Welcker | Sales |
| … | … | … | … |

## Employee Departments

Write a query that selects:

* **EmployeeID**
* **FirstName**
* **Salary**
* **DepartmentName**

Filter only **employees** with **salary higher than 15000**. Return the **first 5** rows **sorted** by **DepartmentID** in **ascending** order.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **Salary** | **DepartmentName** |
| 3 | Roberto | 43300.00 | Engineering |
| 9 | Gail | 32700.00 | Engineering |
| … | … | … | … |

## Employees Without Project

Write a query that selects:

* **EmployeeID**
* **FirstName**

Filter only **employees** **without** a **project**. Return the **first 3** rows **sorted** by **EmployeeID** in **ascending** order.

### Example:

|  |  |
| --- | --- |
| **EmployeeID** | **FirstName** |
| 2 | Kevin |
| 6 | David |
| … | … |

## Employees Hired After

Write a query that selects:

* **FirstName**
* **LastName**
* **HireDate**
* **DeptName**

Filter only **employees** **hired after 1.1.1999** and are from either **"Sales"** or **"Finance"** departments, s**orted** by **HireDate** (**ascending**).

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstName** | **LastName** | **HireDate** | **DeptName** |
| Debora | Poe | 2001-01-19 00:00:00 | Finance |
| Wendy | Kahn | 2001-01-26 00:00:00 | Finance |
| … | … | … | … |

## Create View Highest Peak

Write a SQL query to create view **v\_HighestPeak that selects** all the information about the **highest peak** in the table Peaks. Use the **Geography** **database**.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **PeakName** | **Elevation** | **MountainId** |
| 68 | Everest | 8848 | 9 |

## Students and Classes

Create **database** called **School**.

Graphical user interface, application

Description automatically generated

The school has **classes** and **students** and each class has **many** students and each student has **many** classes. There should be no student enrolled twice in a course. To create appropriate database you will need:

Table Student columns:

* StudentID – int, identity and primary key
* Name – string with size up to 100

Table Class columns:

* ClassID – int, identity and primary key
* Course – string with size up to 100

Table StudentClassRelation columns:

* StudentID – int and not null
* ClassID – int and not null
* Two FOREIGN KEY with references to tables Student and Class
* Primary key pair of (StudentID, ClassID)

**Insert** the following **data**:

* Add two **students** with names: Olaf Alfonso and Clark Davis
* Add the following **classes**: Biology, Chemistry, Physics, English, Computer Science, History
* The **student** Olaf Alfonso studies in these **classes**: Chemistry, English, History
* The **student** Clark Davis studies in these **classes**: Biology, Physics, History

The table StudentClassRelation should look like this:

### Example:

|  |  |
| --- | --- |
| **StudentID** | **ClassID** |
| 1 | 2 |
| 1 | 4 |
| 1 | 6 |
| 2 | 1 |
| 2 | 3 |
| 2 | 6 |