# Упражнение: БД и езикът SQL

## Показване цялата информация за отделите

Напишете SQL заявка, за да намерите **всичката налична информация за отделите**.

### Пример

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

### Подсказка

Graphical user interface, table

Description automatically generated

## Показване всички имена на департаменти

Напишете SQL заявка, за да намерите **всички имена на отдели**.

### Пример

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

### Подсказка

Graphical user interface, application

Description automatically generated

## Заплата на всеки служител

Напишете SQL заявка, която извлича **име**, **фамилия** и **заплата** за всеки служител.

### Пример

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

## Всички различни заплати на служители

Напишете SQL заявка, която намира **всички различни заплати на служители**. Покажете само заплатите.

### Пример

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

## Имена на всички служители по заплата в даден диапазон

Напишете SQL заявка, за да намерите **име**, **фамилия** и **длъжност** на всички служители, чиято **заплата е в интервала [20000, 30000].**

### Пример

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

## Всички служители без мениджър

Напишете SQL заявка, която намира **име и фамилия** на тези служители, които са **без мениджър.**

### Пример

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

## Всички служители със заплата повече от 50000

Напишете SQL заявка, за да намерите **име**, **фамилия** и **заплата** на тези служители, които са със заплата над 50000. Подредете ги по заплата, низходящо.

### Пример

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

## %-те най – добре платени служители

Напишете SQL заявка, която намира **име и фамилия** на **5-те най – добре платени служители** подредени **в низходящ ред на заплатите им.**

### Пример

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

## 7-те последно наети служители

Напишете SQL заявка, която намира **последните 7 наети служители**. Изберете **име, фамилия и дата на наемане на работа**.

### Пример

|  |  |  |
| --- | --- | --- |
| **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 |
| … | … | … |

## Увеличение на заплати

Напишете SQL заявка, която увеличава заплатите на всички служители, които са в отделите: **Engineering**, **Tool Design**, **Marketing** или **Information Services** с **12%**. После **изберете колоната Salaries** от таблицата **Employees**.

### Пример

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

## Адрес на служител

Напишете заявка, която избира:

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

Върнете **първите** **5** реда, **сортирани** по **AddressId** във **възходящ** ред.

### Пример:

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

## Адреси с градове

Напишете заявка, която избира:

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

**Сортирани** по **FirstName** във **възходящ** ред, после по **LastName**. изберете **първите 50** служители.

### Пример:

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

## Служители отдел „Продажби“

Напишете заявка, която избира

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

**Сортирани по** **EmployeeID** във **възходящ** ред. Изберете само **служители** от отдел “**Sales**”.

### Пример:

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

## Отдели

Напишете заявка, която избира:

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

Филтрирайте само **служители** със **заплата по -висока от 15000**. Върнете **първите 5** реда **сортирани** по **DepartmentID** във **възходящ** ред.

### Пример:

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

## Служители без проекти

Напишете заявка, която избира:

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

Филтрирайте само **служители без проекти**. Върнете **първите 3** реда **сортирани** по **EmployeeID** във **възходящ** ред.

### Пример:

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

## Служители наети след определена дата

Напишете заявка, която избира:

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

Филтрирайте само **служители наети след 1.1.1999** , които са или от отдел **"Sales"** или от **"Finance“**, сортирани по **HireDate** (**възходящо**).

### Пример:

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

## Създайте изглед Highest Peak

Напишете SQL заявка, която създава изглед **v\_HighestPeak, който избира** всичката информация за **най – високия връх** в таблица Peaks. Използвайте БД **Geography**.

### Пример:

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

## Ученици и класове

Създайте **БД** наречена **School**.

Graphical user interface, application

Description automatically generated

Училището има **класове** и **ученици** и всеки клас има **много** ученици и всеки ученик има **много** класове. Не бива да има ученик, включен два пъти в някой клас. За да изградите подходящата БД имате нужда от:

Таблица Student, с колони:

* StudentID – int, identity и първичен ключ
* Name – string с размер 100

Таблица Class, с колони:

* ClassID – int, identity и първичен ключ
* Course – string с размер 100

Таблица StudentClassRelation, колони:

* StudentID – int and not null
* ClassID – int and not null
* Два ВЪНШНИ КЛЮЧА с референции към таблиците Student и Class
* Първичен ключ, комбинация от 2 колони (StudentID, ClassID)

**Вмъкнете** следните **данни**:

* Добавете двама **ученици** с имена: Olaf Alfonso и Clark Davis
* Добавете следните **класове**: Biology, Chemistry, Physics, English, Computer Science, History
* **Ученикът** Olaf Alfonso учи в следните **класове**: Chemistry, English, History
* **Ученикът** Clark Davis учи в следните **класове**: Biology, Physics, History

Таблицата StudentClassRelation трябва да изглежда така:

### Пример:

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