**175. Combine Two Tables**

**Easy**

**2.6K**

**204**

**Companies**

**SQL Schema**

**Table: Person**

**+-------------+---------+**

**| Column Name | Type |**

**+-------------+---------+**

**| personId | int |**

**| lastName | varchar |**

**| firstName | varchar |**

**+-------------+---------+**

**personId is the primary key column for this table.**

**This table contains information about the ID of some persons and their first and last names.**

**Table: Address**

**+-------------+---------+**

**| Column Name | Type |**

**+-------------+---------+**

**| addressId | int |**

**| personId | int |**

**| city | varchar |**

**| state | varchar |**

**+-------------+---------+**

**addressId is the primary key column for this table.**

**Each row of this table contains information about the city and state of one person with ID = PersonId.**

**Write an SQL query to report the first name, last name, city, and state of each person in the Person table. If the address of a personId is not present in the Address table, report null instead.**

**Return the result table in any order.**

**The query result format is in the following example.**

**Example 1:**

**Input:**

**Person table:**

**+----------+----------+-----------+**

**| personId | lastName | firstName |**

**+----------+----------+-----------+**

**| 1 | Wang | Allen |**

**| 2 | Alice | Bob |**

**+----------+----------+-----------+**

**Address table:**

**+-----------+----------+---------------+------------+**

**| addressId | personId | city | state |**

**+-----------+----------+---------------+------------+**

**| 1 | 2 | New York City | New York |**

**| 2 | 3 | Leetcode | California |**

**+-----------+----------+---------------+------------+**

**Output:**

**+-----------+----------+---------------+----------+**

**| firstName | lastName | city | state |**

**+-----------+----------+---------------+----------+**

**| Allen | Wang | Null | Null |**

**| Bob | Alice | New York City | New York |**

**+-----------+----------+---------------+----------+**

**Explanation:**

**There is no address in the address table for the personId = 1 so we return null in their city and state.**

**addressId = 1 contains information about the address of personId = 2.**

**Ans:**

create database practice\_new;

use practice\_new;

create table Person(

personId int(10),

lastName varchar(50),

firstName varchar(50),

PRIMARY KEY(personId));

create table Address(

addressId int(20),

personId int(20),

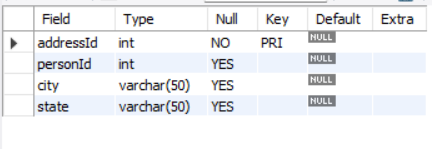
city varchar(50),

state varchar(50),

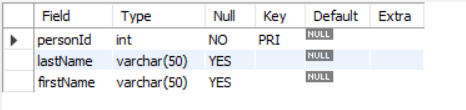
PRIMARY KEY (addressId));

show tables;

desc address;



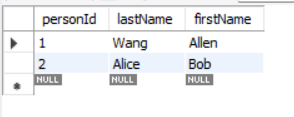
desc person;



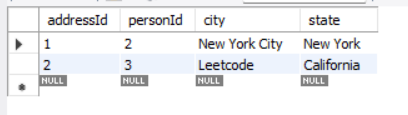
insert into person (personId, lastname, firstname) values (1,'Wang','Allen'),(2,'Alice','Bob');

insert into Address (addressId, personId, city, state) values (1,2,'New York City','New York'),(2,3,'Leetcode','California');

select \* from person;



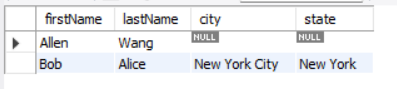
select \* from Address;



Select firstName, lastName, city, state from person p

left join Address a

on p.personId=a.personId;

****

**182. Duplicate Emails**

**Easy**

**1.5K**

**53**

**Companies**

**SQL Schema**

**Table: Person**

**+-------------+---------+**

**| Column Name | Type |**

**+-------------+---------+**

**| id | int |**

**| email | varchar |**

**+-------------+---------+**

**id is the primary key column for this table.**

**Each row of this table contains an email. The emails will not contain uppercase letters.**

**Write an SQL query to report all the duplicate emails.**

**Return the result table in any order.**

**The query result format is in the following example.**

**Example 1:**

**Input:**

**Person table:**

**+----+---------+**

**| id | email |**

**+----+---------+**

**| 1 | a@b.com |**

**| 2 | c@d.com |**

**| 3 | a@b.com |**

**+----+---------+**

**Output:**

**+---------+**

**| Email |**

**+---------+**

**| a@b.com |**

**+---------+**

**Explanation: a@b.com is repeated two times.**

**Ans:**

DROP TABLE IF EXISTS NewPerson;

create TABLE NewPerson (

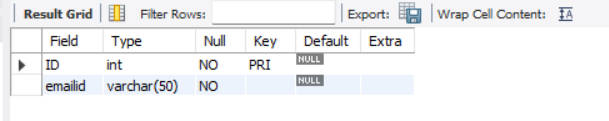
ID int(11) NOT NULL,

emailid varchar(50) NOT NULL,

PRIMARY KEY (ID)

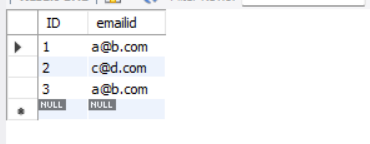
);

desc NewPerson;



insert into NewPerson values(1,'a@b.com'),(2,'c@d.com'),(3,'a@b.com');

Select \* from NewPerson;



SELECT emailid , COUNT(emailid) as cnt

FROM NewPerson

GROUP BY emailid

HAVING COUNT(emailid) > 1;



**185. Department Top Three Salaries**

**Hard**

**1.4K**

**197**

**Companies**

**SQL Schema**

**Table: Employee**

**+--------------+---------+**

**| Column Name | Type |**

**+--------------+---------+**

**| id | int |**

**| name | varchar |**

**| salary | int |**

**| departmentId | int |**

**+--------------+---------+**

**id is the primary key column for this table.**

**departmentId is a foreign key of the ID from the Department table.**

**Each row of this table indicates the ID, name, and salary of an employee. It also contains the ID of their department.**

**Table: Department**

**+-------------+---------+**

**| Column Name | Type |**

**+-------------+---------+**

**| id | int |**

**| name | varchar |**

**+-------------+---------+**

**id is the primary key column for this table.**

**Each row of this table indicates the ID of a department and its name.**

**A company's executives are interested in seeing who earns the most money in each of the company's departments. A high earner in a department is an employee who has a salary in the top three unique salaries for that department.**

**Write an SQL query to find the employees who are high earners in each of the departments.**

**Return the result table in any order.**

**The query result format is in the following example.**

**Example 1:**

**Input:**

**Employee table:**

**+----+-------+--------+--------------+**

**| id | name | salary | departmentId |**

**+----+-------+--------+--------------+**

**| 1 | Joe | 85000 | 1 |**

**| 2 | Henry | 80000 | 2 |**

**| 3 | Sam | 60000 | 2 |**

**| 4 | Max | 90000 | 1 |**

**| 5 | Janet | 69000 | 1 |**

**| 6 | Randy | 85000 | 1 |**

**| 7 | Will | 70000 | 1 |**

**+----+-------+--------+--------------+**

**Department table:**

**+----+-------+**

**| id | name |**

**+----+-------+**

**| 1 | IT |**

**| 2 | Sales |**

**+----+-------+**

**Output:**

**+------------+----------+--------+**

**| Department | Employee | Salary |**

**+------------+----------+--------+**

**| IT | Max | 90000 |**

**| IT | Joe | 85000 |**

**| IT | Randy | 85000 |**

**| IT | Will | 70000 |**

**| Sales | Henry | 80000 |**

**| Sales | Sam | 60000 |**

**+------------+----------+--------+**

**Explanation:**

**In the IT department:**

**- Max earns the highest unique salary**

**- Both Randy and Joe earn the second-highest unique salary**

**- Will earns the third-highest unique salary**

**In the Sales department:**

**- Henry earns the highest salary**

**- Sam earns the second-highest salary**

**- There is no third-highest salary as there are only two employees**

**Ans:**

DROP TABLE IF EXISTS Employee;

create TABLE Employee (

ID int(11) NOT NULL,

Name varchar(50) NOT NULL,

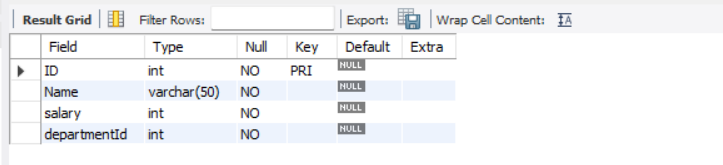
salary int(50) NOT NULL,

departmentId int(50) NOT NULL,

PRIMARY KEY (ID)

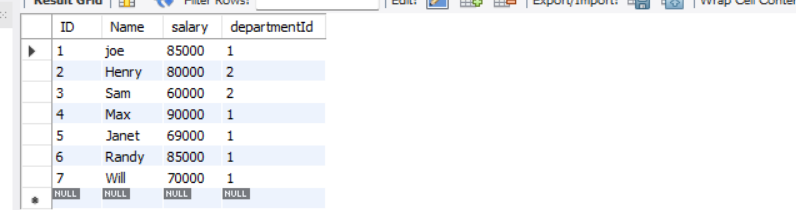
);

desc Employee;

****

insert into Employee values(1,'joe',85000,1),(2,'Henry ',80000,2),(3,'Sam',60000,2),(4,'Max',90000 ,1),(5,'Janet',69000,1),(6,'Randy',85000,1),(7,'Will',70000,1);

Select \* from Employee;



DROP TABLE IF EXISTS Department;

create TABLE Department (

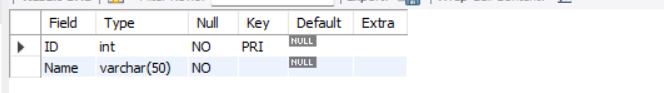
ID int(11) NOT NULL,

Name varchar(50) NOT NULL,

PRIMARY KEY (ID)

);

desc Department;



insert into Department values(1,'IT'),(2,'Sales');

Select \* from Department;



Select d.Name,e.Name, e.Salary from Employee e

left join Department d

on d.ID=e.departmentId

order by Salary desc;

