**Online Resources:**

**Video**

<https://www.youtube.com/watch?v=HXV3zeQKqGY>

**Web Pages**

<https://sqlbolt.com/>

<https://www.w3resource.com/sql-exercises/employee-database-exercise/index.php>

**3 Days**

1. <https://www.w3resource.com/sql-exercises/employee-database-exercise/index.php>

Write a query to find the grade and name of the salary-grade wise employee getting the highest salary.

**Query**

SELECT emp.emp\_name AS highest\_salary\_employee,

g.grade

FROM employees emp

JOIN (SELECT f.grade,

Max(f.salary)AS Sal

FROM (SELECT \*

FROM salary\_grade

CROSS JOIN (SELECT \*

FROM employees) e) f

WHERE f.salary > f.min\_sal

AND f.salary < f.max\_sal

GROUP BY f.grade) g

ON emp.salary = g.sal

ORDER BY g.grade ASC

1. Consider following Table

|  |  |  |  |
| --- | --- | --- | --- |
| Id | date | amount | deposit\_withdrawal |
| 1 | 10-03-2021 | 1000 | deposit |
| 2 | 15-03-2021 | 2000 | deposit |
| 3 | 20-04-2021 | 100 | withdrawal |
| 4 | 22-04-2021 | 300 | deposit |
| 5 | 30-06-2021 | 1500 | withdrawal |

Write a query to produce following result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | date | deposit | withdrawal | balance |
| 1 | 10-03-2021 | 1000 |  | 1000 |
| 2 | 15-03-2021 | 2000 |  | 3000 |
| 3 | 20-04-2021 |  | 100 | 2900 |
| 4 | 22-04-2021 | 300 |  | 3200 |
| 5 | 30-06-2021 |  | 1500 | 1700 |

**Query**

SELECT ID, TRANSACTION\_DATE,DEPOSIT, WITHDRAW,

(BALANCE\_D - BALANCE\_W) AS BALANCE

FROM(

SELECT

ID,TRANSACTION\_DATE,DEPOSIT,WITHDRAW,

SUM (DEPOSIT) OVER(ORDER BY TRANSACTION\_DATE) AS BALANCE\_D,

SUM (WITHDRAW) OVER(ORDER BY TRANSACTION\_DATE) AS BALANCE\_W

FROM (

SELECT

ID,TRANSACTION\_DATE,

CASE

WHEN DEPOSIT\_WITHDRAWAL ='DEPOSIT' THEN AMOUNT ELSE 0

END AS DEPOSIT,

CASE

WHEN DEPOSIT\_WITHDRAWAL ='WITHDRAWAL' THEN AMOUNT ELSE 0

END AS WITHDRAW,

FROM TRANSACTION )

GROUP BY TRANSACTION\_DATE)

1. Create a table and put the following contents in the table. Assuming the quantity of each book sold per record is 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Book** | **Price** | **Customer** | **Date** |
| Arthur Conan Doyle | A Study in Scarlet | $50 | Perry mason | 26/11/2020 |
| Agatha Christie | The Murder of Roger Ackroyd | $52 | Erle Stanley Gardner | 25/11/2020 |
| Rabindranath Tagore | Gora | $52 | Miss Marple | 25/11/2020 |
| Agatha Christie | Murder on the Orient Express | $14 | Miss Marple | 28/11/2020 |
| Agatha Christie | Murder on the Orient Express | $14 | Mark Twain | 24/11/2020 |
| Mark Twain | The Adventures of Tom Sawyer | $26 | Agatha Christie | 24/11/2020 |
| Agatha Christie | The Murder of Roger Ackroyd | $52 | Arthur Conan Doyle | 26/11/2020 |
| Erle Stanley Gardner | Careless Kitten | $15 | Agatha Christie | 28/11/2020 |
| Mark Twain | The Gilded Age | $24 | Perry Mason | 26/11/2020 |
| Rabindranath Tagore | Gora | $52 | Miss Marple | 28/11/2020 |
| Mark Twain | The Gilded Age | $24 | Perry Mason | 25/11/2020 |
| Agatha Christie | Peril at End House | $53 | Arthur Conan Doyle | 28/11/2020 |
| Mark Twain | The Adventures of Tom Sawyer | $26 | Agatha Christie | 28/11/2020 |
| Mark Twain | The Gilded Age | $24 | Rabindranath Tagore | 24/11/2020 |
| Rabindranath Tagore | Gitanjali | $41 | Mark Twain | 24/11/2020 |
| Agatha Christie | Peril at End House | $53 | Mark Twain | 25/11/2020 |
| Mark Twain | The Adventures of Huckleberry Finn | $27 | Arthur Conan Doyle | 27/11/2020 |
| Arthur Conan Doyle | The Sign of Four | $26 | Harry Potter | 26/11/2020 |
| Erle Stanley Gardner | Careless Kitten | $15 | Sherlock Holms | 27/11/2020 |
| Rabindranath Tagore | Gitanjali | $40 | Perry Mason | 24/11/2020 |
| Rabindranath Tagore | Gitanjali | $50 | Miss Marple | 28/11/2020 |
| Rabindranath Tagore | Gitanjali | $50 | Perry Mason | 28/11/2020 |

Write a query to find book-wise

no of distinct customers who are authors, (count1)

no of distinct customers who are not authors. (count2)

Sample Output

**BookName count1 count2**

.

.

.

Gitanjali 1 2

**QUERY**

**SELECT a.book AS bookName,**

**a.count1,**

**b.count2**

**FROM (SELECT Count (DISTINCT customer) AS count1,**

**book**

**FROM book\_record**

**WHERE customer IN (SELECT author**

**FROM book\_record)**

**GROUP BY book) a**

**JOIN (SELECT Count (DISTINCT customer) AS count2,**

**book**

**FROM book\_record**

**WHERE customer NOT IN (SELECT author**

**FROM book\_record)**

**GROUP BY book) b**

**ON a.book = b.book**

.

1. Write a query to find author wise total cost of books sold in descending order by total cost.

**QUERY**

**SELECT author,**

**Sum(price) AS cost**

**FROM book\_record**

**GROUP BY author**

**ORDER BY cost DESC**

1. Find the name of books having the second highest cost.

**QUERY**

**SELECT book**

**FROM (SELECT book,**

**Sum(price) AS cost**

**FROM book\_record**

**GROUP BY book)**

**WHERE cost < (SELECT Max(cost)**

**FROM (SELECT book,**

**Sum(price) AS cost**

**FROM book\_record**

**GROUP BY book))**

**ORDER BY cost DESC**

**LIMIT 1**

1. Find the date on which the highest amount received.

**QUERY**

**SELECT date**

**FROM (SELECT date,**

**Sum(price) AS cost**

**FROM book\_record**

**GROUP BY date)**

**WHERE cost = (SELECT Max(cost) AS cost**

**FROM (SELECT date,**

**Sum(price) AS cost**

**FROM book\_record**

**GROUP BY date))**