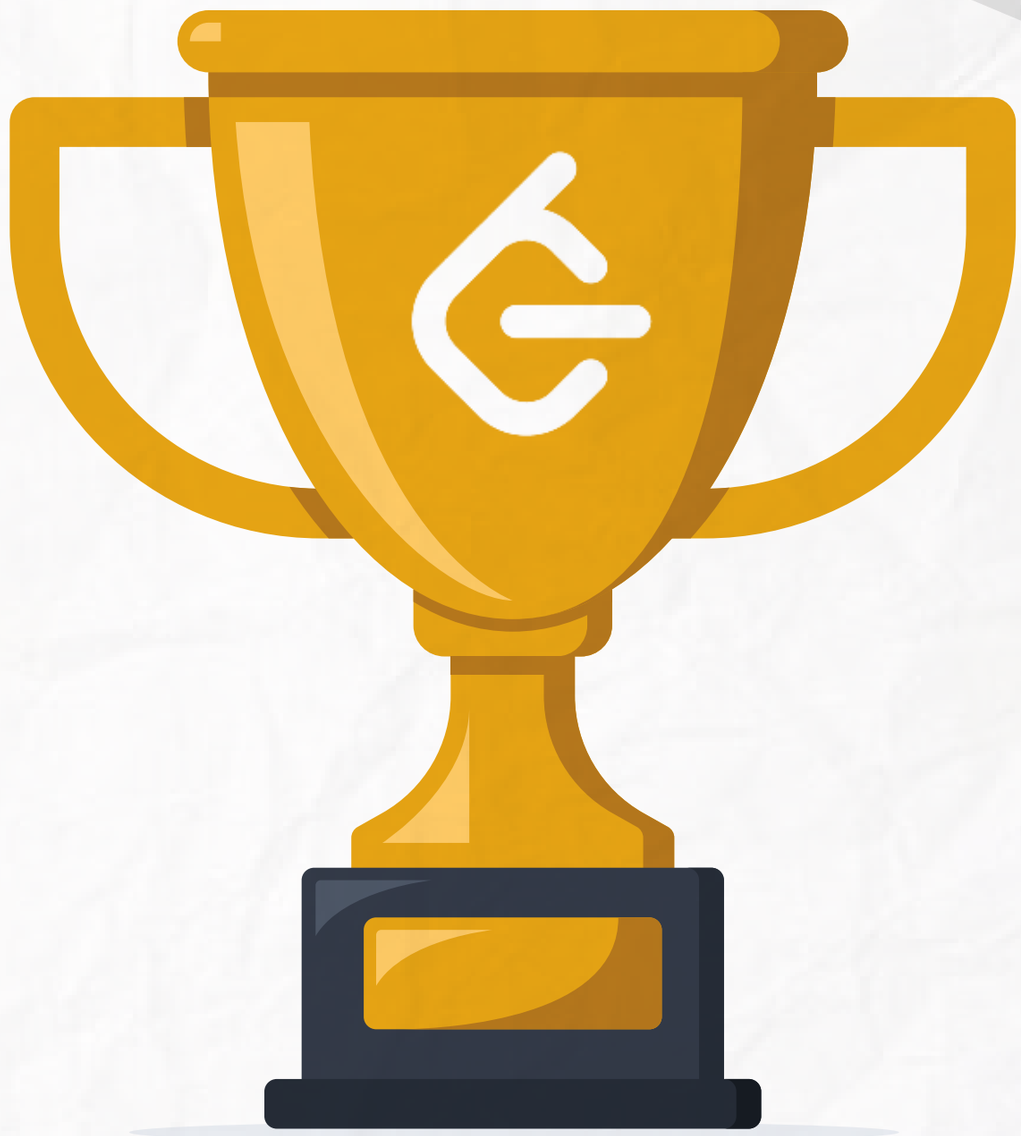
BOSSCODER ACADEMY

LeetCode

SQL 50

Challenge



Disclaimer

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.

SELECT

PROBLEM 01

Recyclable and Low Fat Products

Table: `Products`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| product_id  | int     |
| low_fats    | enum    |
| recyclable  | enum    |
+-------------+---------+
product_id is the primary key (column with unique values) for this table.
low_fats is an ENUM (category) of type ('Y', 'N') where 'Y' means this product is
low fat and 'N' means it is not.
recyclable is an ENUM (category) of types ('Y', 'N') where 'Y' means this product
is recyclable and 'N' means it is not.
```

Write a solution to find the ids of products that are both low fat and recyclable. Return the result table in any order. The result format is in the following example.

Practice Here →

PROBLEM 02

Find Customer Referee

Table: `Customer`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| id          | int     |
| name        | varchar |
| referee_id  | int     |
+-------------+---------+
In SQL, id is the primary key column for this table.
Each row of this table indicates the id of a customer, their name, and the id of
the customer who referred them.
```

Find the names of the customers that are **not referred by** the customer with `id = 2`. Return the result table in **any order**.

The result format is in the following example.

Practice Here →

PROBLEM 03

Big Countries

Table: `World`

Column Name	Type
name	varchar
continent	varchar
area	int
population	int
gdp	bigint

name is the primary key (column with unique values) for this table.
Each row of this table gives information about the name of a country, the continent to which it belongs, its area, the population, and its GDP value.

A country is big if:
- it has an area of at least three million (i.e., `3000000 km2`), or
- it has a population of at least twenty-five million (i.e., `25000000`).

Write a solution to find the name, population, and area of the big countries. Return the result table in any order. The result format is in the following example.

Practice Here →

PROBLEM 04

Article Views I

Table: `Views`

Column Name	Type
article_id	int
author_id	int
viewer_id	int
view_date	date

There is no primary key (column with unique values) for this table, the table may have duplicate rows.
Each row of this table indicates that some viewer viewed an article (written by some author) on some date.
Note that equal author_id and viewer_id indicate the same person.

Write a solution to find all the authors who viewed at least one of their own articles. Return the result table sorted by `id` in ascending order. The result format is in the following example.

Practice Here →

PROBLEM 05

Invalid Tweets

Table: `Tweets`

Column Name	Type
tweet_id	int
content	varchar

tweet_id is the primary key (column with unique values) for this table.
This table contains all the tweets in a social media app.

Write a solution to find the IDs of the invalid tweets. The tweet is invalid if the number of characters used in the content of the tweet is strictly greater than 15.

Return the result table in any order.

The result format is in the following example.

Practice Here →

BASIC JOINS

PROBLEM 06

Replace Employee ID With The Unique Identifier

Table: `Employees`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| id            | int     |
| name          | varchar |
+---------------+---------+
id is the primary key (column with unique values) for this table.
Each row of this table contains the id and the name of an employee in a company.
```

Table: `EmployeeUNI`

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| id            | int     |
| unique_id     | int     |
+---------------+---------+
(id, unique_id) is the primary key (combination of columns with unique values)
for this table.
Each row of this table contains the id and the corresponding unique id of an
employee in the company.
```

Write a solution to show the **unique ID** of each user, If a user does not have a unique ID replace just show `null`.
Return the result table in **any** order.
The result format is in the following example.

Practice Here →

PROBLEM 07

Product Sales Analysis I

Table: `Sales`

Column Name	Type
sale_id	int
product_id	int
year	int
quantity	int
price	int

(sale_id, year) is the primary key (combination of columns with unique values) of this table.
product_id is a foreign key (reference column) to `Product` table.
Each row of this table shows a sale on the product product_id in a certain year.
Note that the price is per unit.

Table: `Product`

Column Name	Type
product_id	int
product_name	varchar

product_id is the primary key (column with unique values) of this table.
Each row of this table indicates the product name of each product.

Write a solution to report the `product_name`, `year`, and `price` for each `sale_id` in the `Sales` table.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 08

Customer Who Visited but Did Not Make Any Transactions

Table: `Visits`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| visit_id    | int     |
| customer_id | int     |
+-------------+---------+
visit_id is the column with unique values for this table.
This table contains information about the customers who visited the mall.
```

Table: `Transactions`

```
+----------------+---------+
| Column Name    | Type    |
+----------------+---------+
| transaction_id | int     |
| visit_id       | int     |
| amount         | int     |
+----------------+---------+
transaction_id is column with unique values for this table.
This table contains information about the transactions made during the visit_id.
```

Write a solution to find the IDs of the users who visited without making any transactions and the number of times they made these types of visits.
Return the result table sorted in any order.
The result format is in the following example.

Practice Here →

PROBLEM 09

Rising Temperature

Table: `Weather`

Column Name	Type
id	int
recordDate	date
temperature	int

id is the column with unique values for this table.
There are no different rows with the same recordDate.
This table contains information about the temperature on a certain day.

Write a solution to find all dates' `Id` with higher temperatures compared to its previous dates (yesterday).

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 10

Average Time of Process per Machine

Table: `Activity`

Column Name	Type
machine_id	int
process_id	int
activity_type	enum
timestamp	float

The table shows the user activities for a factory website.
(machine_id, process_id, activity_type) is the primary key (combination of columns with unique values) of this table.
machine_id is the ID of a machine.
process_id is the ID of a process running on the machine with ID machine_id.
activity_type is an ENUM (category) of type ('start', 'end').
timestamp is a float representing the current time in seconds.
'start' means the machine starts the process at the given timestamp and 'end' means the machine ends the process at the given timestamp.
The 'start' timestamp will always be before the 'end' timestamp for every (machine_id, process_id) pair.

There is a factory website that has several machines each running the same number of processes. Write a solution to find the average time each machine takes to complete a process.

The time to complete a process is the 'end' timestamp minus the 'start' timestamp. The average time is calculated by the total time to complete every process on the machine divided by the number of processes that were run.

The resulting table should have the machine_id along with the average time as processing_time, which should be rounded to 3 decimal places.

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 11

Employee Bonus

Table: `Employee`

Column Name	Type
empId	int
name	varchar
supervisor	int
salary	int

empId is the column with unique values for this table.
Each row of this table indicates the name and the ID of an employee in addition to their salary and the id of their manager.

Table: `Bonus`

Column Name	Type
empId	int
bonus	int

empId is the column of unique values for this table.
empId is a foreign key (reference column) to empId from the Employee table.
Each row of this table contains the id of an employee and their respective bonus.

Write a solution to report the name and bonus amount of each employee with a bonus less than 1000.

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 12

Student and Examinations

Table: `Students`

Column Name	Type
student_id	int
student_name	varchar

student_id is the primary key (column with unique values) for this table.
Each row of this table contains the ID and the name of one student in the school.

Table: `Subjects`

Column Name	Type
subject_name	varchar

subject_name is the primary key (column with unique values) for this table.
Each row of this table contains the name of one subject in the school.

Write a solution to find the number of times each student attended each exam.

Return the result table ordered by `student_id` and `subject_name`.

The result format is in the following example.

Practice Here →

PROBLEM 13

Managers with at Least 5 Direct Reports

Table: `Employee`

Column Name	Type
id	int
name	varchar
department	varchar
managerId	int

id is the primary key (column with unique values) for this table.
Each row of this table indicates the name of an employee, their department, and the id of their manager.
If managerId is null, then the employee does not have a manager.
No employee will be the manager of themself.

Write a solution to find managers with at least five direct reports.

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 14

Confirmation Rate

Table: `Signups`

Column Name	Type
user_id	int
time_stamp	datetime

user_id is the column of unique values for this table.
Each row contains information about the signup time for the user with ID user_id.

Table: `Confirmations`

Column Name	Type
user_id	int
time_stamp	datetime
action	ENUM

(user_id, time_stamp) is the primary key (combination of columns with unique values) for this table.
user_id is a foreign key (reference column) to the Signups table.
action is an ENUM (category) of the type ('confirmed', 'timeout')
Each row of this table indicates that the user with ID user_id requested a confirmation message at time_stamp and that confirmation message was either confirmed ('confirmed') or expired without confirming ('timeout').

The confirmation rate of a user is the number of '`confirmed`' messages divided by the total number of requested confirmation messages. The confirmation rate of a user that did not request any confirmation messages is `0`. Round the confirmation rate to two decimal places.
Write a solution to find the confirmation rate of each user.
Return the result table in any order.
The result format is in the following example.

Practice Here →

BASIC AGGREGATE FUNCTIONS

PROBLEM 15

Not Boring Movies

Table: `Cinema`

```
+----------------+----------+
| Column Name    | Type     |
+----------------+----------+
| id             | int      |
| movie          | varchar  |
| description    | varchar  |
| rating         | float    |
+----------------+----------+
id is the primary key (column with unique values) for this table.
Each row contains information about the name of a movie, its genre, and its
rating.
rating is a 2 decimal places float in the range [0, 10]
```

Write a solution to report the movies with an odd-numbered ID and a description that is not "`boring`".
Return the result table ordered by `rating` in descending order.
The result format is in the following example.

Practice Here →

PROBLEM 16

Confirmation Rate

Table: `Users`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| user_id     | int     |
| user_name   | varchar |
+-------------+---------+
user_id is the primary key (column with unique values) for this table.
Each row of this table contains the name and the id of a user.
```

Table: `Register`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| contest_id  | int     |
| user_id     | int     |
+-------------+---------+
(contest_id, user_id) is the primary key (combination of columns with unique
values) for this table.
Each row of this table contains the id of a user and the contest they registered
into.
```

Write a solution to find the percentage of the users registered in each contest rounded to two decimals.

Return the result table ordered by `percentage` in descending order. In case of a tie, order it by `contest_id` in ascending order.

The result format is in the following example.

Practice Here →

PROBLEM 17

Project Employees I

Table: `Project`

Column Name	Type
project_id	int
employee_id	int

(project_id, employee_id) is the primary key of this table.
employee_id is a foreign key to `Employee` table.
Each row of this table indicates that the employee with employee_id is working on the project with project_id.

Table: `Employee`

Column Name	Type
employee_id	int
name	varchar
experience_years	int

employee_id is the primary key of this table. It's guaranteed that experience_years is not NULL.
Each row of this table contains information about one employee.

Write an SQL query that reports the average experience years of all the employees for each project, rounded to 2 digits.

Return the result table in any order.

The query result format is in the following example.

Practice Here →

PROBLEM 18

Average Selling Price

Table: `Prices`

Column Name	Type
product_id	int
start_date	date
end_date	date
price	int

(product_id, start_date, end_date) is the primary key (combination of columns with unique values) for this table.
Each row of this table indicates the price of the product_id in the period from start_date to end_date.
For each product_id there will be no two overlapping periods. That means there will be no two intersecting periods for the same product_id.

Write a solution to find the average selling price for each product. `average_price` should be rounded to 2 decimal places.

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 19

Percentage of Users Attended a Contest

Table: `Users`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| user_id     | int     |
| user_name   | varchar |
+-------------+---------+
user_id is the primary key (column with unique values) for this table.
Each row of this table contains the name and the id of a user.
```

Table: `Register`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| contest_id  | int     |
| user_id     | int     |
+-------------+---------+
(contest_id, user_id) is the primary key (combination of columns with unique
values) for this table.
Each row of this table contains the id of a user and the contest they registered
into.
```

Write a solution to find the percentage of the users registered in each contest rounded to two decimals.

Return the result table ordered by `percentage` in descending order. In case of a tie, order it by `contest_id` in ascending order.
The result format is in the following example.

Practice Here →

PROBLEM 20

Queries Quality and Percentage

Table: `Queries`

Column Name	Type
query_name	varchar
result	varchar
position	int
rating	int

This table may have duplicate rows.
This table contains information collected from some queries on a database.
The `position` column has a value from **1** to **500**.
The `rating` column has a value from **1** to **5**. Query with `rating` less than 3 is a poor query.

We define query `quality` as:
The average of the ratio between query rating and its position.
We also define `poor query percentage` as:
The percentage of all queries with rating less than 3.
Write a solution to find each `query_name`,the `quality` and `poor_query_percentage`.
Both `quality` and `poor_query_percentage` should be rounded to 2 decimal places.
Return the result table in any order.

Practice Here →

PROBLEM 21

Monthly Transaction I

Table: `Transactions`

Column Name	Type
id	int
country	varchar
state	enum
amount	int
trans_date	date

id is the primary key of this table.
The table has information about incoming transactions.
The state column is an enum of type ["approved", "declined"].

Write an SQL query to find for each month and country, the number of transactions and their total amount, the number of approved transactions and their total amount.

Return the result table in any order.

The query result format is in the following example.

Practice Here →

PROBLEM 22

Immediate Food Delivery II

Table: `Delivery`

Column Name	Type
delivery_id	int
customer_id	int
order_date	date
customer_pref_delivery_date	date

delivery_id is the column of unique values of this table.
The table holds information about food delivery to customers that make orders at some date and specify a preferred delivery date (on the same order date or after it).

If the customer's preferred delivery date is the same as the order date, then the order is called immediate; otherwise, it is called scheduled.
The first order of a customer is the order with the earliest order date that the customer made. It is guaranteed that a customer has precisely one first order.
Write a solution to find the percentage of immediate orders in the first orders of all customers, rounded to 2 decimal places.
The result format is in the following example.

Practice Here →

PROBLEM 23

Game Play Analysis IV

Table: `Activity`

Column Name	Type
player_id	int
device_id	int
event_date	date
games_played	int

(player_id, event_date) is the primary key (combination of columns with unique values) of this table.
This table shows the activity of players of some games.
Each row is a record of a player who logged in and played a number of games (possibly 0) before logging out on someday using some device.

Write a solution to report the fraction of players that logged in again on the day after the day they first logged in, rounded to 2 decimal places. In other words, you need to count the number of players that logged in for at least two consecutive days starting from their first login date, then divide that number by the total number of players.

The result format is in the following example.

Practice Here →

SORTING AND GROUPING

PROBLEM 24

Number of Unique Subjects Taught by Each Teacher

Table: `Teacher`

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| teacher_id  | int  |
| subject_id  | int  |
| dept_id     | int  |
+-------------+------+
(subject_id, dept_id) is the primary key (combinations of columns with unique values) of this table.
Each row in this table indicates that the teacher with teacher_id teaches the subject subject_id in the department dept_id.
```

Write a solution to calculate the number of unique subjects each teacher teaches in the university.
Return the result table in any order.
The result format is shown in the following example.

Practice Here →

PROBLEM 25

User Activity for the Past 30 Days I

Table: `Activity`

Column Name	Type
user_id	int
session_id	int
activity_date	date
activity_type	enum

This table may have duplicate rows.
The activity_type column is an ENUM (category) of type ('open_session', 'end_session', 'scroll_down', 'send_message').
The table shows the user activities for a social media website.
Note that each session belongs to exactly one user.

Write a solution to find the daily active user count for a period of 30 days ending 2019-07-27 inclusively. A user was active on someday if they made at least one activity on that day.

Return the result table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 26

Product Sales Analysis III

Table: `Sales`

Column Name	Type
sale_id	int
product_id	int
year	int
quantity	int
price	int

(sale_id, year) is the primary key (combination of columns with unique values) of this table.
product_id is a foreign key (reference column) to `Product` table.
Each row of this table shows a sale on the product product_id in a certain year.
Note that the price is per unit.

Table: `Product`

Column Name	Type
product_id	int
product_name	varchar

product_id is the primary key (column with unique values) of this table.
Each row of this table indicates the product name of each product.

Write a solution to select the `product_id`, year, quantity, and price for the first year of every product sold.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 27

Classes More Than 5 Students

Table: `Courses`

Column Name	Type
student	varchar
class	varchar

(student, class) is the primary key (combination of columns with unique values) for this table.
Each row of this table indicates the name of a student and the class in which they are enrolled.

Write a solution to select the `product_id`, year, quantity, and price for the first year of every product sold.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 28

Find Followers Count

Table: `Followers`

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| user_id     | int  |
| follower_id | int  |
+-------------+------+
(user_id, follower_id) is the primary key (combination of columns with unique
values) for this table.
This table contains the IDs of a user and a follower in a social media app where
the follower follows the user.
```

Write a solution to select the `product id`, year, quantity, and price for the first year of every product sold.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 29

Biggest Single Number

Table: `MyNumbers`

```
+-------------+------+
| Column Name | Type |
+-------------+------+
| num         | int  |
+-------------+------+
This table may contain duplicates (In other words, there is no primary key for
this table in SQL).
Each row of this table contains an integer.
```

Write a solution to select the `product id`, year, quantity, and price for the first year of every product sold.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 30

Customers Who Bought All Products

Table: `Customer`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| customer_id | int     |
| product_key | int     |
+-------------+---------+
```

This table may contain duplicates rows.
`customer_id` is not NULL.
product_key is a foreign key (reference column) to `Product` table.

Table: `Product`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| product_key | int     |
+-------------+---------+
```

product_key is the primary key (column with unique values) for this table.

Write a solution to select the `product_id`, year, quantity, and price for the first year of every product sold.

Return the resulting table in any order.

The result format is in the following example.

Practice Here →

PROBLEM 31

Customers Who Bought All Products

Table: `Customer`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| customer_id | int     |
| product_key | int     |
+-------------+---------+
```

This table may contain duplicates rows.
`customer_id` is not NULL.
product_key is a foreign key (reference column) to `Product` table.

Table: `Product`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| product_key | int     |
+-------------+---------+
```

product_key is the primary key (column with unique values) for this table.

Write a solution to calculate the number of bank accounts for each salary category. The salary categories are:
"`Low Salary`": All the salaries **strictly less** than `$20000`.
"`Average Salary`": All the salaries in the **inclusive** range [`$20000`, `$50000`].
"`High Salary`": All the salaries **strictly greater** than `$50000`.
The result table **must** contain all three categories. If there are no accounts in a category, return `0`.
Return the result table in **any order**.
The result format is in the following example.

Practice Here →

ADVANCED SELECT AND JOINS

PROBLEM 32

The Number of Employees Which Report to Each Employee

Practice Here →

PROBLEM 33

Primary Department for Each Employee

Practice Here →

PROBLEM 34

Triangle Judgement

Practice Here →

PROBLEM 35

Consecutive Numbers

Practice Here →

PROBLEM 36

Product Price at a Given Date

Practice Here →

PROBLEM 37

Last Person to Fit in the Bus

Practice Here →

PROBLEM 38

Count Salary Categories

Practice Here →

SUBQUERIES

PROBLEM 39

Employees Whose Manager Left the Company

Practice Here →

PROBLEM 40

Exchange Seats

Practice Here →

PROBLEM 41

Movie Rating

Practice Here →

PROBLEM 42

Restaurant Growth

Practice Here →

PROBLEM 43

Friend Requests II: Who Has the Most Friends

Practice Here →

PROBLEM 44

Investments in 2016

Practice Here →

PROBLEM 45

Department Top Three Salaries

Practice Here →

ADVANCED STRING FUNCTIONS / REGEX / CLAUSE

PROBLEM 46

Fix Names in a Table

Practice Here →

PROBLEM 47

Patients With a Condition

Practice Here →

PROBLEM 48

Delete Duplicate Emails

Practice Here →

PROBLEM 49

Second Highest Salary

Practice Here →

PROBLEM 50

Group Sold Products By The Date

Practice Here →

BOSSCODER ACADEMY

WHY BOSSCODER?

750+ Alumni placed at Top Product-based companies.

More than **136% hike** for every **2 out of 3** working professional.

Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company

Rahul .
Google

EXPLORE MORE