PROapp DATABASE DESIGN

Ngoc Ngan (Zoel) Huynh

BUSA8090 - Data and Visualisation for Business

Table of Contents

INTRODUCTION

This report is prepared for the Chief Operating Officer (COO) of PROapp to provide valuable insights into the management of newly registered users on our online platform. In response to the COO's request to minimise overhead and enhance operational efficiency, we leverage the robust database architecture designed for our platform. This database supports user management and facilitates a deeper understanding of user behaviour and trends. Furthermore, the report outlines nine potential challenges related to user engagement, payment patterns, certification requirements, and other critical factors. Each issue is accompanied by meticulously formulated SQL queries designed to extract relevant data, allowing us to back our insights with concrete evidence. Hence, the following sections will elaborate on the database structure, detail the rationale behind the selected queries, and present the anticipated results.

PROAPP BUSINESS BRIEF

The PROapp is a sharing economy platform designed to connect taskers and customers for a variety of services, including but not limited to home repairs, tutoring, gardening, and more. By leveraging the skills of certified taskers and the needs of customers, PROapp aims to create a user-friendly environment that fosters trust and efficiency in service delivery.

Key objectives:

1. User engagement: Attract and retain a diverse pool of customers and taskers through effective marketing and engagement strategies.
2. Certification and training: Enhance the quality of service by ensuring taskers have the necessary certifications and training to perform tasks competently.
3. Revenue generation: Establish multiple revenue streams through transaction fees, membership fees, and quote fees while maintaining competitive pricing for users.
4. Data-driven decision-making: Utilise data analytics to identify trends, improve service offerings, and streamline operations based on user behaviour and feedback.
5. Geographical expansion: Explore opportunities for expansion into new markets by analysing demand and supply dynamics at regional and local levels.

Major challenges:

1. Managing newly registered users: With a growing number of users, effective data management practices are essential to minimise overhead and enhance user experience.
2. Balancing demand and supply: Understanding the availability of taskers about customer demand is crucial for operational efficiency.
3. Quality assurance: Ensuring taskers meet quality standards through certification while providing customers with a reliable selection of services.
4. Revenue sustainability: Establishing a consistent revenue model that accommodates varying user needs and preferences.

Main entities:

1. Customer:
    - A customer can post multiple tasks.
    - New customers may not yet post any task.

- Key information stored: name, email, phone number, country, city, date of becoming a customer and a brief description about his/herself.

2. Tasker:
   - A Tasker performs tasks and must be categorised as either:
     - Supplier: Companies that provide specialised services (e.g., builders, landscapers).
     - Tradesperson: Individuals certified in trades (e.g., plumbers, electricians).
   - Taskers can bid on multiple tasks, and new Taskers may have no bids yet.
   - Some taskers need to register for membership to take certifications for specific tasks.
   - One task can be bid by multiple taskers.
   - Key information stored: name, email, phone number, country, city, date of becoming a tasker, which type of tasker he/she is, and a brief description about his/herself.
3. Certification:
   - Tracks taskers' progress and training for each certification, including checks and training results.
   - A tasker can take multiple certifications in different training fields or none, and a certification can be taken by multiple taskers or none.
   - Information stored: Certification ID, certification name, estimated time to complete (in hours), training field, expire after how many years.
   - If the certification has no expiration date, the field “Expire_After” should be left null.
4. Membership:
   - This entity tracks the membership status of taskers.
   - Only taskers with an active membership can participate in certifications. Without membership, taskers are ineligible for any certifications.
     - Active Membership: Required for all certifications.
     - Expired Membership: Taskers with expired memberships cannot pursue new certifications but retain previously obtained certifications.
     - If the membership expires while the tasker is mid-training, the certification process cannot be completed until the membership is renewed.

- A tasker may register for membership multiple times. If the tasker's membership expires while they are mid-training, they must renew their membership to continue and complete the certification process.
- Membership must belong to only one tasker.
- Information stored: membership type (monthly/yearly), registration date, expiry date, and membership fee.

5. Training and assessment:
    - This associative entity tracks the training and assessment process taskers undergo to obtain certifications. To be qualified for certification, the tasker must pass:
        - Background check: Verifies education, experience, etc.
        - Police check: Ensures no criminal history.
        - Code of practice test: Confirms knowledge of the trade's standards.
        - Training result: Pass
    - Information stored: background check status, police check status, code of practice check status, training result (pass/fail), and date recorded for each entry.
6. Task:
    - A task is posted by only one customer and can receive multiple bids or none from taskers.
    - Information stored: task title, budget, task description, due date, creation date, certification requirement (if applicable), and area of expertise.
7. Task assignment:
    - This associative entity links taskers and tasks and includes information about the bid, review, and rating.
    - Taskers can bid on multiple tasks, and new Taskers may have no bids yet.
    - After changing the task status to "Completed," both the tasker and customer must leave the rating corresponding to the task. The task review is optional.
    - Information stored: bid price, date when inserting new records, customer and tasker ratings (scale of 5) and reviews, task status, bid status (Successful/ Pending/ Failed), cancellation reason (if applicable), and cancelled by (customer/ tasker) (if applicable).

- Task status flow:
    - “Not assigned” (when the bid status is pending)
    - "Assigned" (after the bid is accepted by the customer)
    - "Completed" (after confirmation from both parties)
    - "Cancelled" (if the task is withdrawn)

8. Payment:

- This associative entity links taskers and customers to represent the payment transaction.
- Customers pay the tasker for tasks they have completed.
- One customer can pay multiple taskers or none if they do not have any task completed.
- One tasker can receive multiple payments or none.
- Information stored: payment transaction timestamp, payment amount, payment type (Credit Card, PayPal, E-money), and payment status (Pass/Failed).

This part is essential as it serves as the foundation for the data analytics report, guiding the analysis of user data and identification of organisational issues that may impact the COO’s decision-making process. Through a detailed examination of the current database and strategic use of SQL queries, the report aims to provide actionable insights that align with the overarching objectives of PROapp.

PROCESS OF CONSTRUCTING DATABASE

1. Create an entity relationship diagram (ERD)

To create an ERD for PROapp, we need to follow a series of steps:

Step 1: Identify the entities

From the business case, we can extract the following entities:

1. Customer
2. Tasker
3. Certification
4. Membership

5. Training and assessment
6. Task
7. Task assignment
8. Payment

Step 2: Identify business rules:

1. Customer posts task.
2. Tasker bids on task.
3. The tasker is involved in task assignments.
4. Task belongs to task assignments.
5. Tasker registers for membership to take certifications.
6. Tasker takes certification.
7. The tasker takes part in training and assessment.
8. Training and assessment involves certification.
9. Customer pays tasker.
10. The tasker is involved in payment.
11. The customer is involved in payment.

Step 3: Define relationships and represent cardinality and participation

Associative entities:

1. Task assignment: linked between task and tasker
2. Training and assessment: linked between tasker and certification
3. Payment: linked between customer and tasker

Cardinality:

1. Customer - Task (One-to-Many)
    - One customer can post multiple tasks.
    - A task is posted by only one customer.
2. Tasker - Task (Many-to-Many)
    - A tasker can bid on multiple tasks.
    - A task can receive multiple bids from taskers.
3. Tasker - Certification (Many-to-Many)
    - A tasker can take multiple certifications.
    - A certification can be taken by multiple taskers.
4. Tasker - Membership (One-to-Many)
    - A tasker can register for membership multiple times.
    - A membership belongs to only one tasker.
5. Customer – Tasker (Many-to-Many)

- A customer can pay multiple taskers.
- A tasker can be paid by multiple customers.

Participation:

1. Customer - Task
   - Some new customers may not post any task (Partial).
   - A task is posted by only one customer (Total).
2. Tasker - Task
   - Some new taskers may not bid on any task (Partial).
   - Some tasks may not receive any bids from taskers (Partial).
3. Tasker - Certification
   - Some taskers may not take any certifications (Partial).
   - Some certifications may not be taken by any taskers (Partial).
4. Tasker - Membership
   - Some taskers may not register for membership (Partial).
   - A membership must belong to a tasker (Total).
5. Customer – Tasker (Many-to-Many)
   - Some customers may not pay any taskers (Partial).
   - Some taskers may not be paid by any customers (Partial).

Step 4: Identify attributes

1. Customer
   - customer_id (Primary key): Unique identifier for each customer.
   - name: Full name of the customer.
   - email: Email address of the customer for communication and notifications.
   - phone_number: Contact number for reaching the customer.
   - country: Country where the customer resides.
   - city: City where the customer resides.
   - registration_date: Date when the customer registered on the platform.
   - description: Brief description about the customer (optional).
2. Tasker
   - tasker_id (primary key): Unique identifier for each tasker.
   - name: Full name of the tasker.
   - email: Email address for communication.
   - phone_number: Contact number for reaching the tasker.
   - country: Country where the tasker is located.
   - city: City where the tasker is located.
   - registration_date: Date when the tasker registered on the platform.

- tasker_type: Type of tasker (either Supplier or Tradesperson).
- description: Brief description about the tasker (optional).

3. Certification
   - certification_id (primary key): Unique identifier for each certification.
   - certification_name: Name of the certification.
   - estimated_completion_time: The estimated time required to complete the certification (in hours).
   - training_field: Area of expertise related to the certification.
   - expire_after_years: Number of years until the certification expires; nullable if there is no expiration.
4. Membership
   - membership_id (primary key): Unique identifier for each membership record.
   - membership_type: Type of membership (Monthly or Yearly).
   - registration_date: Date when the membership was registered.
   - expiry_date: Date when the membership expires.
   - membership_fee: Fee associated with the membership.
5. Training and assessment (Certification_Tasker)
   - tasker_id (primary key): References the tasker undergoing assessment.
   - certification_id (primary key): References the certification undergoing assessment.
   - date_recorded (primary key): Date when the assessment entry was recorded.

   => Composite primary key to identify each record: tasker_id, certification_id, date_recorded
   - background_check_status: Status of the background check (e.g., Pass/Fail).
   - police_check_status: Status of the police check (e.g., Pass/Fail).
   - code_of_practice_status: Status of the code of practice check (e.g., Pass/Fail).
   - training_result: Result of the training (Pass/Fail).
6. Task
   - task_id (primary key): Unique identifier for each task.
   - task_title: Title of the task.
   - budget: The budget set by the customer for the task.
   - task_description: Detailed description of the task.
   - due_date: Deadline for task completion.
   - creation_date: Date when the task was created.
   - certification_required: Indicates whether certification is required for the task (Boolean).
   - expertise_area: Area of expertise required for the task.
7. Task assignment (Tasker_Task)
   - task_id (primary key): References the task associated with the assignment.

- tasker_id (primary key): References the tasker associated with the assignment.
- date_recorded (primary key): Date when the assignment entry was recorded.

=> Composite primary key to identify each record: tasker_id, task_id, date_recorded

- bid_price: Price bid by the tasker for the task.
- customer_rating: Rating given by the customer (scale of 1 to 5).
- tasker_rating: Rating given by the tasker (scale of 1 to 5).
- customer_review: Review given by the customer (optional)
- tasker_review: Review given by the tasker (optional)
- task_status: Status of the task (Not assigned, Assigned, Completed, Cancelled).
- bid_status: Status of the bid (Successful, Pending, Failed).
- cancellation_reason: Reason for cancellation (optional).
- cancelled_by: Indicates who cancelled the task (Customer/Tasker, optional).

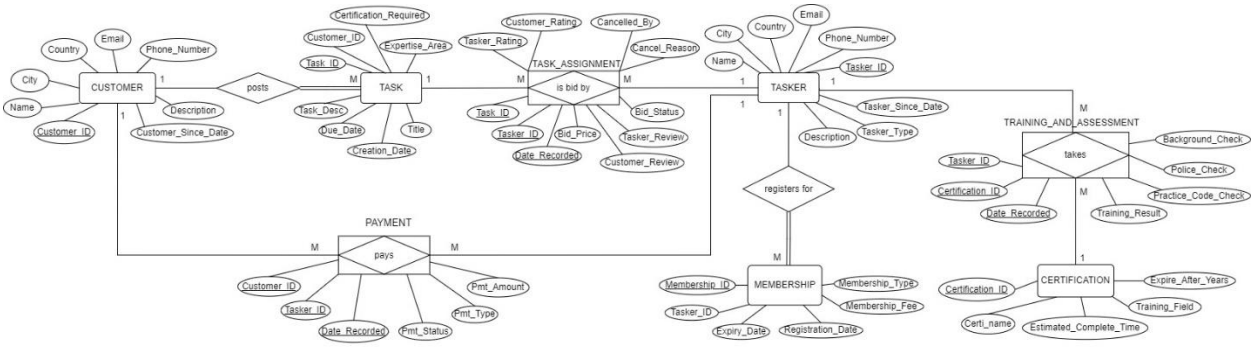
8. Payment (Customer_Tasker)

- customer_id (primary key): References the customer making the payment.
- tasker_id (primary key): References the tasker receiving the payment.
- date_recorded (primary key): Date and time of the payment transaction.

=> Composite primary key to identify each record: tasker_id, customer_id, date_recorded

- payment_amount: Amount paid by the customer.
- payment_type: Method of payment (Credit Card, PayPal, E-money).
- payment_status: Status of the payment transaction (Pass, Failed).

Result:



Please access this link for a higher-resolution image: PROapp_ERD

2. Database design in MySQL

This step is done in MySQL.

First, we use data definition language to:

- Create tables for each entity
- Create relationships between tables

Example codes (the full version is available in the PART A folder submitted along with this file):

```
-- Table 6: Task
CREATE TABLE Task (
    task_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT NOT NULL,
    task_title VARCHAR(255),
    budget DECIMAL(10, 2),
    task_desc TEXT,
    due_date DATETIME,
    creation_date DATETIME,
    certification_required BOOLEAN,
    expertise_area VARCHAR(255),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);
```

The next step is inserting the data into our tables. In this case, the bulk insert method and MySQL Import Tool are utilised to facilitate the process.

Example codes (CSV data files are available in the PART A folder submitted along with this file):

```
-- Insert task values
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Task.csv"
INTO TABLE task
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(customer_id,task_title,budget,task_desc,due_date,creation_date,certification_required,expertise_area);
-- check if inserting is valid
-- select * from task;

-- Import task_assignment values through MYSQL Import Tool
-- check if inserting is valid
-- select * from task_assignment;
```

Example result of task table (using bulk insert method):

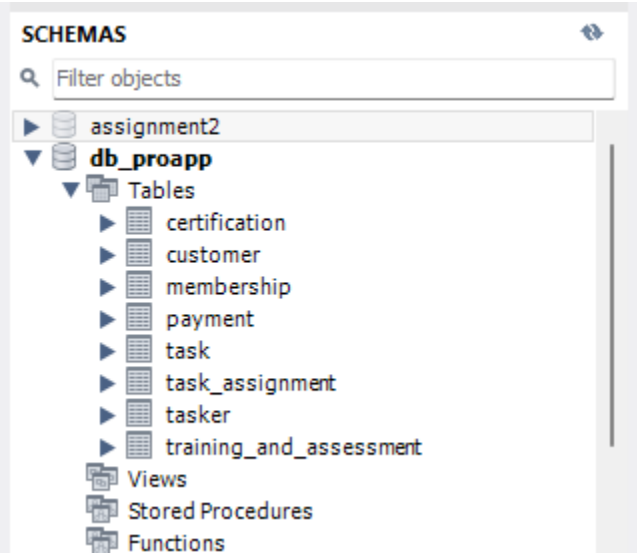
task_id	customer_id	task_title	budget	task_desc	due_date	creation_date	certification_required	expertise_area
1	12	Build a garden shed	1500.00	Construct a garden shed in my backyard.	2010-05-24 12:00:00	2026-09-23 08:00:00	1	Landscaping
2	45	Install new plumbing	2000.00	Need to replace old pipes in the kitchen.	2015-03-24 09:00:00	2020-09-23 09:00:00	1	Plumbing
3	67	Electricity upgrade	1800.00	Upgrade electrical system for safety.	2030-01-24 14:00:00	2015-09-23 12:00:00	1	Electrical
4	23	House painting	1200.00	Paint the interior of the house.	2010-02-24 10:00:00	2018-09-23 11:00:00	0	Painting
5	38	Create a backyard patio	2500.00	Install a stone patio in my backyard.	2025-04-24 13:30:00	2022-09-23 15:00:00	1	Landscaping
6	92	Garden landscaping	900.00	Design and implement a garden layout.	2005-06-24 10:30:00	2030-09-23 10:30:00	1	Landscaping
7	54	Home electrical repair	400.00	Fix flickering lights in living room.	2015-10-23 08:00:00	2021-09-23 10:00:00	0	Electrical
8	33	Install security system	700.00	Need help installing a new security system.	2001-11-23 14:00:00	2023-09-23 15:00:00	1	Electrical
9	19	New roof installation	5000.00	Replace the entire roof of the house.	2001-04-24 12:00:00	2014-09-23 09:00:00	1	Construction
10	76	Clean gutters	250.00	Need someone to clean my gutters.	2020-10-23 16:00:00	2017-09-23 10:00:00	0	Cleaning
11	1	Install new windows	3000.00	Replace old windows with new ones.	2001-03-24 10:00:00	2028-09-23 10:00:00	1	Construction
12	68	Make a custom shelf	500.00	Build a shelf for my living room.	2010-01-24 11:00:00	2029-09-23 09:00:00	0	Carpentry
13	99	Design an office space	800.00	Need help designing my home office.	2015-02-24 13:00:00	2011-09-23 08:00:00	0	Interior Design
14	20	Carpet installation	1200.00	Install new carpet in living room.	2025-10-23 10:30:00	2016-09-23 15:00:00	1	Flooring
15	12	Build a fence	1500.00	Construct a wooden fence around my yard.	2015-05-24 12:00:00	2026-09-23 08:00:00	1	Construction
16	45	Install bathroom tiles	1000.00	Need tiles installed in my bathroom.	2020-03-24 09:00:00	2020-09-23 09:00:00	1	Tiling

Example result of task assignment table (using MySQL Import Tool):

task_id	tasker_id	bid_price	date_recorded	customer_rating	tasker_rating	task_status	bid_status	cancel_reason	cancelled_by
1	5	150.00	2015-09-23 09:00:00	5	4	Assigned	Successful	NULL	NULL
1	10	120.00	2016-09-23 09:00:00	NULL	NULL	Not assigned	Pending	NULL	NULL
1	15	130.00	2017-09-23 09:00:00	NULL	NULL	Not assigned	Failed	NULL	NULL
2	3	200.00	2018-09-23 10:00:00	4	5	Completed	Successful	NULL	NULL
2	8	210.00	2019-09-23 10:30:00	5	5	Completed	Successful	NULL	NULL
2	20	190.00	2020-09-23 11:00:00	NULL	NULL	Not assigned	Pending	NULL	NULL
3	17	240.00	2022-09-23 12:30:00	5	5	Assigned	Successful	NULL	NULL
3	20	250.00	2021-09-23 12:00:00	NULL	NULL	Not assigned	Failed	NULL	NULL
3	25	230.00	2023-09-23 13:00:00	NULL	NULL	Not assigned	Pending	NULL	NULL
4	11	290.00	2025-09-23 14:30:00	5	4	Assigned	Successful	NULL	NULL
4	25	300.00	2024-09-23 14:00:00	3	3	Not assigned	Failed	Tasker unava...	Tasker
5	7	175.00	2026-09-23 15:00:00	NULL	NULL	Not assigned	Failed	NULL	NULL
5	19	185.00	2027-09-23 15:30:00	5	5	Assigned	Successful	NULL	NULL

The rest of the tables are created, and their data is inserted in the same way.

By doing so, we have the final database created for PROapp.



POTENTIAL ORGANISATIONAL ISSUES

Potential Problem 1: Imbalance of Demand and Supply Across Regions

An imbalance in demand and supply across regions can hinder growth and affect customer satisfaction. If marketing resources are misallocated, the company may experience underperformance in key markets, ultimately leading to wasted expenditures and missed revenue opportunities. Hence, some steps need to be implemented:

Step 1: Calculate the total taskers and customers

```
-- Total taskers and customers on the platform
SELECT tasker_stats.total_taskers, customer_stats.total_customers
FROM (
    SELECT COUNT(tasker_id) AS total_taskers
    FROM tasker
) AS tasker_stats
JOIN (
    SELECT COUNT(customer_id) AS total_customers
    FROM customer
) AS customer_stats;
```

Query elaboration: This query is formulated to help the COO gain an overview of the platform's user base by calculating the total number of taskers and customers. The inner queries count the total taskers and customers separately, and the results are joined to provide a snapshot of supply (taskers) vs. demand (customers). Understanding this distribution will allow for data-driven decisions regarding resource allocation, marketing efforts, and potential adjustments to balance supply and demand.

Suggested return result:

	total_taskers	total_customers
▶	250	100

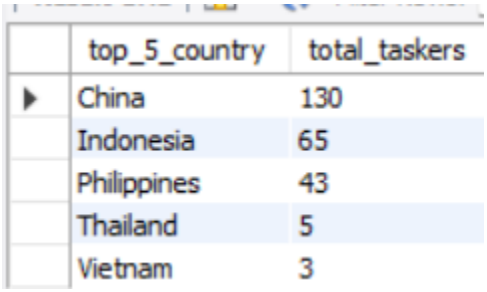
Recommendation: The current supply of taskers exceeds customer demand, indicating the need to focus marketing efforts on attracting more customers to the marketplace.

Step 2: Identify the top 5 regions with the most taskers

```sql
-- Top 5 countries by tasker count
SELECT country as top_5_country, COUNT(tasker_id) AS total_taskers
FROM tasker
GROUP BY country
ORDER BY total_taskers DESC
LIMIT 5;
```

Query elaboration: This query focuses on identifying the regions with the highest number of taskers. By grouping taskers by country and sorting by the total count, the query returns the top 5 countries with the highest supply (taskers). This information helps in understanding geographic supply concentrations, allowing the COO to tailor regional marketing efforts or tasker acquisition strategies.

Suggested return result:



top_5_country	total_taskers
China	130
Indonesia	65
Philippines	43
Thailand	5
Vietnam	3

Recommendation: The top 5 countries with the highest supply sources have been identified. Our strategy will focus on aligning this supply with the corresponding demand in these regions.

Step 3: Identify the top 5 regions with the most customers

```sql
-- Top 5 countries by customer count
SELECT country as top_5_country, COUNT(customer_id) AS total_customers
FROM customer
GROUP BY country
ORDER BY total_customers DESC
LIMIT 5;
```

Query elaboration: This query mirrors the logic of the tasker query but focuses on the demand side, i.e., customers. By identifying the top 5 countries where customer counts are highest, the COO can compare this data against the tasker distribution to identify regions with mismatched supply and demand. This insight can lead to better decisions around customer acquisition or balancing supply in those areas.

Suggested return result:

top_5_country	total_customers
China	18
Indonesia	12
Russia	8
Brazil	6
Philippines	5

Recommendation: Similar to step 2, this data highlights areas with high demand concentration. While China and Indonesia continue to lead, new markets such as Russia, Brazil, and the Philippines are emerging. Based on this insight, we should prioritize investment in the Chinese and Indonesian markets.

Potential Problem 2: Insufficient Engagement Rate on The Platform

Low engagement rates indicate that registered users are not fully utilising the platform, resulting in higher administrative costs and lost opportunities. Addressing this can lead to better resource allocation and increased revenue potential.

```
SELECT COUNT(DISTINCT t.tasker_id) AS active_taskers,
       t_stat.total_taskers,
       COUNT(DISTINCT c.customer_id) AS active_customers,
       c_stat.total_customers
FROM customer c
JOIN task tsk ON c.customer_id = tsk.customer_id
JOIN task_assignment ta ON tsk.task_id = ta.task_id
JOIN tasker t ON ta.tasker_id = t.tasker_id
JOIN (SELECT COUNT(*) AS total_taskers FROM tasker) AS t_stat
JOIN (SELECT COUNT(*) AS total_customers FROM customer) AS c_stat;
```

Query elaboration: This query helps measure the engagement rate of taskers and customers by comparing the total registered users to those who are actively involved in tasks or bids. By joining task assignments with taskers and customers, the query identifies those who are actively participating, while the subqueries count the total number of users. This comparison highlights the active vs. total user ratio, giving the COO insights into platform usage and areas for improvement in engagement.

Suggested return result:

	active_taskers	total_taskers	active_customers	total_customers
▶	21	250	24	100

Recommendation: Engagement rates for both taskers and customers are notably low, with only 21 out of 250 taskers and 24 out of 100 customers actively participating. This suggests potential underlying issues, such as a non-user-friendly interface or a lack of diverse job opportunities and taskers, that may be hindering engagement.

Potential Problem 3: Lack of Understanding of Customer Spending Behaviour

A lack of insight into customer spending behaviour can impede effective pricing strategies and marketing efforts. Understanding spending patterns enables the identification of high-demand expertise areas, allowing for tailored offerings and revenue strategies.

```
SELECT expertise_area, AVG(budget) AS avg_budget, COUNT(tsk.task_id) AS task_count
FROM task tsk
GROUP BY expertise_area
ORDER BY avg_budget DESC;
```

Query elaboration: This query is designed to analyse customer spending behaviour by focusing on the average budget spent per expertise area. Grouping tasks by expertise area and calculating the average budget provides insights into which fields customers are willing to spend more on. Understanding this behaviour enables better revenue strategy planning, helping the COO allocate resources and set pricing policies that align with customer willingness to pay.

Suggested return result:

	expertise_area	avg_budget	task_count
▶	Construction	3430.000000	20
	HVAC	2437.500000	4
	Interior Design	1871.428571	7
	Painting	1600.000000	2
	Electrical	1553.846154	13
	Flooring	1466.666667	3

Recommendation: Customers show a strong willingness to spend on construction-related tasks, which also have high demand. To leverage this, we can increase transaction and bid fee rates for

these tasks, fostering a more competitive environment that attracts high-quality taskers while also boosting revenue.

Potential Problem 4: Insufficient Revenue and the Need for Expanding Revenue Sources

Reliance on limited revenue streams can risk financial stability. Therefore, analysing revenue sources is crucial to uncovering potential growth areas and optimising existing revenue mechanisms.

```
SELECT 'Transaction Fee' AS source, SUM(p.pmt_amount) * 0.1 AS total_revenue -- every transaction costs 10% to the app
FROM Payment p
WHERE p.pmt_status = 'Successful'
UNION ALL
SELECT 'Membership Fee', SUM(m.membership_fee) AS total_revenue
FROM Membership m
UNION ALL
SELECT 'Quote Fee', SUM(ta.bid_price) * 0.01 -- assuming that every successful bid is charged 1% of bid price, and this is
FROM Task_Assignment ta
WHERE bid_status = 'Successful'
ORDER BY total_revenue DESC;
```

Query elaboration: This query examines the three main revenue streams: transaction fees, membership fees, and quote fees. The query uses 'union all' to combine the results from each revenue source, with appropriate calculations for each type (10% of successful payments for transaction fees and 1% of successful bids for quote fees). This breakdown provides insights into which revenue sources are most profitable, helping the COO decide where to focus revenue expansion efforts or adjust pricing strategies.

Suggested return result:

source	total_revenue
Membership Fee	12250.0000
Transaction Fee	2599.0000
Quote Fee	302.4000

Recommendation: Membership fees are currently the most profitable source of revenue, so marketing efforts should prioritize growing the membership base. However, transaction and quote fees are relatively low. To address this, we can consider adjusting the base rates, particularly for high-demand tasks (as mentioned in problem 3), to further boost revenue.

Potential Problem 5: Low Rating and High Rate of Cancellations

High cancellation rates and low ratings negatively impact platform credibility, deterring potential users. Addressing these issues can enhance user experience and attract more customers and taskers.

Step 1: Average ratings, cancellation rate, and bid price

```
SELECT AVG(ta.customer_rating) AS avg_customer_rating,
       AVG(ta.tasker_rating) AS avg_tasker_rating,
       tsk.expertise_area,
       AVG(ta.bid_price) AS avg_bid_price,
       AVG(task_status = 'Cancelled')*100 AS avg_cancellation_rate
FROM task_assignment ta
JOIN task tsk ON ta.task_id = tsk.task_id
GROUP BY tsk.expertise_area
ORDER BY avg_customer_rating ASC, avg_tasker_rating ASC;
```

Query elaboration: This query assesses the performance and satisfaction levels of customers and taskers in various expertise areas. By calculating average ratings and cancellation rates for each area, we can gain insights into which areas are underperforming and may require intervention. Including the average bid price adds a financial dimension, allowing us to analyse whether pricing is correlated with lower ratings or higher cancellations. Our focus is on the low ratings, so the records are sorted in ascending order based on average customer and tasker ratings.

Suggested return result:

avg_customer_rating	avg_tasker_rating	expertise_area	avg_bid_price	avg_cancellation_rate
3.7500	3.5000	Plumbing	225.000000	30.0000
4.0000	3.5000	Painting	260.000000	0.0000
4.0000	3.5000	Tiling	247.500000	0.0000
4.0000	4.0000	Flooring	193.333333	66.6667
4.2667	4.2667	Electrical	212.631579	21.0526
4.4444	4.2444	Carpentry	197.307692	21.1538

Recommendation: Plumbing tasks have the lowest combined customer and tasker ratings, while flooring tasks experience the highest cancellation rates. There appears to be no direct correlation between ratings, bid prices, and cancellation rates. Therefore, we should focus on improving the management of plumbing and flooring tasks. One approach could be introducing penalties, such as fines for taskers who consistently receive low ratings or cancel tasks more than three times consecutively, to enhance marketplace performance.

Step 2: Investigate cancellations

```
-- Deep dive into cancellations
SELECT COUNT(DISTINCT ta.task_id) AS cancelled_tasks, ta.cancel_reason, ta.cancelled_by
FROM Task_Assignment ta
WHERE task_status = 'Cancelled'
GROUP BY ta.cancel_reason, ta.cancelled_by;
```

Query elaboration: This query focuses on understanding why and by whom tasks are being cancelled. By grouping cancellations by reason and the party responsible (customer or tasker), the COO can identify patterns in cancellations and address underlying issues, such as improving communication or setting clearer expectations between customers and taskers.

Suggested return result:

	cancelled_tasks	cancel_reason	cancelled_by
▶	12	Customer changed mind	Customer
	11	Tasker unavailable	Tasker

Recommendation: Task cancellations are equally distributed between customers and taskers, with no discernible pattern in the reasons for cancellations. As a result, no specific recommendation can be made at this time.

Step 3: Identify individual taskers and customers who cancelled the task

```
SELECT c.customer_id, COUNT(distinct ta.task_id) AS cancelled_tasks, ta.cancel_reason, ta.cancelled_by
FROM customer c
JOIN task tsk on c.customer_id = tsk.customer_id
JOIN task_assignment ta on tsk.task_id = ta.task_id
WHERE task_status = 'Cancelled' and ta.cancelled_by = 'Customer'
GROUP BY ta.cancel_reason, ta.cancelled_by, c.customer_id
ORDER BY cancelled_tasks DESC;
```

Query elaboration: This query dives deeper by pinpointing specific customers responsible for task cancellations. Grouping by customer and cancellation reason allows for targeted customer service interventions.

Suggested return result:

	customer_id	cancelled_tasks	cancel_reason	cancelled_by
▶	23	3	Customer changed mind	Customer
	54	3	Customer changed mind	Customer
	68	2	Customer changed mind	Customer
	19	1	Customer changed mind	Customer
	33	1	Customer changed mind	Customer
	45	1	Customer changed mind	Customer

```
SELECT t.tasker_id, COUNT(distinct ta.task_id) AS cancelled_tasks, ta.cancel_reason, ta.cancelled_by
FROM task tsk
JOIN task_assignment ta on tsk.task_id = ta.task_id
JOIN tasker t on ta.tasker_id = t.tasker_id
WHERE task_status = 'Cancelled' and ta.cancelled_by = 'Tasker'
GROUP BY ta.cancel_reason, ta.cancelled_by, t.tasker_id
ORDER BY cancelled_tasks DESC;
```

Query elaboration: A similar query for taskers is executed to identify taskers who frequently cancel tasks, which could be addressed by refining tasker onboarding or training.

Suggested return result:

	tasker_id	cancelled_tasks	cancel_reason	cancelled_by
▶	12	3	Tasker unavailable	Tasker
	4	2	Tasker unavailable	Tasker
	5	2	Tasker unavailable	Tasker
	1	1	Tasker unavailable	Tasker
	6	1	Tasker unavailable	Tasker
	7	1	Tasker unavailable	Tasker

Recommendation: Based on the two lists above, we should implement a penalty system that reduces the credibility of users with poor behaviour. This could involve decreasing job visibility for taskers or limiting the number of tasks customers can post.

Potential Problem 6: Uneven Payment Type Distribution and the High Failure Rates

Understanding payment type distributions and failure rates can reveal inefficiencies and areas for improvement. High failure rates can lead to lost transactions and customer dissatisfaction.

Step 1: Payment type breakdown

```sql
-- Payment trends by type
SELECT p.pmt_type,
       COUNT(*) AS total_payments,
       SUM(p.pmt_amount) AS total_amount
FROM Payment p
GROUP BY p.pmt_type
ORDER BY total_payments DESC;
```

Query elaboration: This query aggregates payment data (payment amount and number of transactions) by type, giving the COO insight into payment trends and user preferences, which can inform marketing strategies to promote preferred payment methods, such as e-money.

Suggested return result:

pmt_type	total_payments	total_amount
Credit Card	56	18920.00
PayPal	31	10500.00
E-money	29	9830.00

Recommendation: Most customers prefer using credit cards for payments, highlighting the need to implement strong security measures for this payment method. However, our core payment option, e-money, is less commonly used. To increase its adoption, we should promote e-money by offering incentives such as discounted transaction fees, to raise awareness and encourage more customers to utilise it.

Step 2: Identify payment types with high failure rates

```sql
-- Failed payments analysis
SELECT pmt_status, ROUND(COUNT(*)*100/(SELECT COUNT(*) FROM Payment),2) as failed_rate, pmt_type
FROM Payment
WHERE pmt_status = 'Failed'
GROUP BY pmt_type;
```

Query elaboration: This query calculates the failure rates for each payment type, providing critical insights into which methods may require additional support or adjustment to improve transaction success rates.

Suggested return result:

	pmt_status	failed_rate	pmt_type
▶	Failed	15.52	PayPal
	Failed	9.48	Credit Card
	Failed	9.48	E-money

Recommendation: Although credit cards are the most used payment method, PayPal has the highest failure rate. This indicates the need to review our PayPal payment system and implement measures to reduce the failure rate.

Potential Problem 7: Limited membership registration

Low membership registration can indicate misalignment between the offerings and tasker needs, hampering user retention and engagement.

Step 1: Compare tasks requiring certification and taskers with memberships

```
-- Tasks requiring certification vs. taskers with membership
SELECT task_stats.total_task_require_certification, membership_stats.total_taskers_have_membership
FROM (
    SELECT COUNT(*) AS total_task_require_certification
    FROM task
    WHERE certification_required = TRUE
) AS task_stats
JOIN (
    SELECT COUNT(distinct tasker_id) AS total_taskers_have_membership
    FROM membership
) AS membership_stats;
```

Query elaboration: By employing subqueries, we isolate counts of tasks that require certification and distinct taskers with membership. We then combined the two results for the direct ratio analysis. This output will inform the COO of potential deficiencies in certified taskers relative to market demand.

Suggested return result:

	total_task_require_certification	total_taskers_have_membership
▶	55	31

Recommendation: There is a mismatch between the number of taskers with memberships and the number of tasks requiring certification. This indicates an opportunity to grow our membership base, and we should increase promotional efforts to target this gap.

Step 2: Investigate membership type distribution

```
-- Membership type distribution
SELECT m.membership_type, COUNT(m.tasker_id) as total_memberships
FROM Membership m
GROUP BY m.membership_type
ORDER BY total_memberships DESC;
```

Query elaboration: The query aggregates the number of taskers grouped by each membership type. This data is crucial for assessing which membership types may require enhanced marketing or adjustments.

Suggested return result:

	membership_type	total_memberships
▶	Monthly	19
	Yearly	13

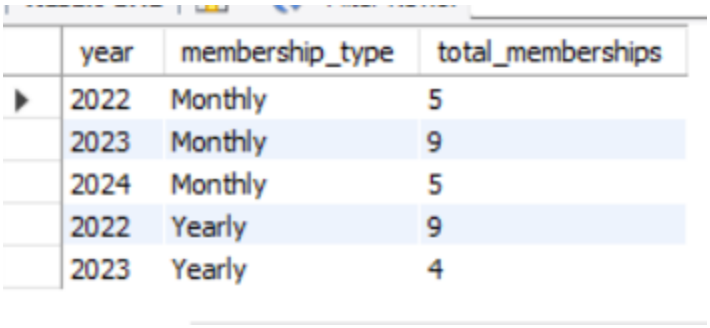
Recommendation: Taskers tend to prefer monthly memberships over yearly ones. However, yearly memberships provide a more stable revenue stream for us and are a strong indicator of higher retention and long-term engagement. To shift this trend, we should consider implementing strategies that incentivize taskers to choose yearly memberships. This could include offering discounted rates for annual commitments, bundling additional benefits exclusive to yearly subscribers, or providing loyalty rewards that make the yearly option more appealing.

Step 3: Analyse membership registration trends over time

```
-- Break down by year
SELECT extract(year from registration_date) as year, membership_type, COUNT(tasker_id) as total_memberships
FROM Membership
GROUP BY year, membership_type
ORDER BY membership_type, year;
```

Query elaboration: This query aims to provide a breakdown of total registrations by each membership type over time, allowing for trend analysis and supporting strategic planning and resource allocation based on observed patterns.

Suggested return result:



year	membership_type	total_memberships
2022	Monthly	5
2023	Monthly	9
2024	Monthly	5
2022	Yearly	9
2023	Yearly	4

Recommendation: The number of monthly memberships fluctuates year to year, while yearly memberships have steadily declined, with none reported in 2024. This suggests that our membership strategy, particularly for yearly bundles, is not performing effectively. To address this, we need to implement targeted methods to improve membership retention and attract more yearly subscribers. These could include enhancing the value proposition of yearly memberships through exclusive benefits, offering more competitive pricing or discounts for long-term commitments, and launching marketing campaigns focused on the advantages of sustained engagement with our platform.

Potential Problem 8: Lack of Certified Taskers for Tasks with Certification

The scarcity of certified taskers could limit the quality and range of services offered, affecting user satisfaction and competitiveness. Hence, some analysis should be done to understand this problem more thoroughly and strategise ways to deal with it.

Step 1: Compare certified taskers and tasks requiring certification

```
-- Tasks requiring certification vs. taskers taking certification
SELECT task_stats.total_task_require_certification, certification_stats.total_taskers_take_certification
FROM (
    SELECT COUNT(*) AS total_task_require_certification
    FROM task
    WHERE certification_required = TRUE
) AS task_stats
JOIN (
    SELECT COUNT(DISTINCT tasker_id) AS total_taskers_take_certification
    FROM training_and_assessment
) AS certification_stats;
```

Query elaboration: This SQL query aggregates data on tasks requiring certification and the number of taskers engaged in certification training. The inner subquery counts tasks flagged as needing certification, while the second subquery counts distinct taskers undergoing training, providing a comprehensive overview of the certification landscape within the platform.

Suggested return result:

	total_task_require_certification	total_taskers_take_certification
▶	55	18

Recommendation: There are more tasks requiring certification than there are certified taskers, highlighting a critical area for improvement. Despite having 19 monthly memberships and 13 yearly memberships, only 18 taskers have completed the certification process. This indicates an issue with our training program or its accessibility. We need to focus on enhancing and promoting our certification process to increase the number of certified taskers, ensuring it better aligns with demand. Strategies could include streamlining the certification process, offering incentives for completion, and improving communication around the benefits of being certified.

Step 2: Identify the top 5 expertise areas requiring certification

```
-- Top 5 expertise area requiring certification
SELECT COUNT(*) AS total_task_require_certification, expertise_area
FROM task
WHERE certification_required = TRUE
GROUP BY expertise_area
ORDER BY total_task_require_certification DESC
LIMIT 5;
```

Query elaboration: This SQL query identifies the top five expertise areas requiring certification by counting tasks marked as needing certification, grouping by expertise area, and ordering results to prioritise critical fields for training. This information is pivotal for determining where to focus training resources and marketing efforts.

Suggested return result:

	total_task_require_certification	expertise_area
▶	16	Construction
	10	Electrical
	7	Plumbing
	5	Landscaping
	4	HVAC

Recommendation: Construction and electrical tasks top the list of those requiring certification. To meet this demand, we should prioritize offering and promoting more certifications in these fields.

By expanding certification opportunities and encouraging taskers to obtain them, we can increase the number of qualified taskers and better match the growing demand in these critical areas.

Step 3: Identify the top 5 training fields taken by taskers

```
-- Top 5 training fields that taskers take the certification
SELECT COUNT(distinct tasker_id) AS total_certifications_taken, training_field
FROM training_and_assessment train_assess
JOIN certification c ON train_assess.certification_id = c.certification_id
GROUP BY training_field
ORDER BY total_certifications_taken DESC;
```

Query elaboration: This SQL query identifies the top five training fields by counting the distinct taskers who have completed certifications. The join between the training_and_assessment and certification table enables a comprehensive count of unique taskers for each training field. The resulting data will inform strategic decisions regarding training programs, highlighting areas where taskers are most interested in pursuing certifications. Additionally, it will help identify any mismatches between the training fields and customer demand. If such mismatches exist, further analysis should be conducted to optimise resource allocation and better align training offerings with market needs.

Suggested return result:

total_certifications_taken	training_field
5	Electrical
5	Health & Safety
5	Plumbing
4	Gardening
3	Childcare

Recommendation: On the other hand, most taskers are opting for certifications in electrical, health & safety, or plumbing, which do not fully align with the areas of highest demand. This trend should not continue, as it creates an imbalance between the supply of certified taskers and the actual market needs. We need to guide taskers towards certifications that better match the demand, such as in construction and other high-need fields.

Potential Problem 9: Low certification pass rates

Low certification pass rates directly impact the quality of taskers available for higher-skilled tasks, potentially indicating underlying issues within the training programs. Addressing these issues is crucial for improving pass rates and ensuring that taskers are well-prepared for their roles.

```
-- Certification pass rates by training field
SELECT c.training_field,
       COUNT(train_assess.certification_id) AS total_certifications,
       SUM(CASE WHEN train_assess.training_result = 'Pass' THEN 1 ELSE 0 END) AS passed_certifications,
       ROUND((SUM(CASE WHEN train_assess.training_result = 'Pass' THEN 1 ELSE 0 END) /
       COUNT(train_assess.certification_id)),2) * 100 AS pass_rate
FROM certification c
JOIN Training_And_Assessment train_assess ON c.certification_id = train_assess.certification_id
GROUP BY c.training_field
ORDER BY pass_rate ASC;
```

Query elaboration: This SQL query evaluates the certification pass rates across various training fields. It counts the total certifications attempted by taskers and the number of those that were successfully passed. By joining the certification and training_and_assessment tables, the query facilitates a detailed analysis of performance across different training fields. The pass rate is calculated as a percentage, providing valuable insights into which training fields exhibit the highest and lowest success rates. This information is instrumental in guiding future improvements in training quality and content, ensuring that taskers receive effective training that aligns with the demands of the market.

Suggested return result:

	training_field	total_certifications	passed_certifications	pass_rate
▶	Plumbing	5	2	40.00
	Health & Safety	5	3	60.00
	Gardening	4	3	75.00
	Electrical	5	5	100.00
	Childcare	3	3	100.00

Recommendation: Plumbing has the lowest certification pass rate. With only 2 out of 5 taskers passing the plumbing certification, it suggests potential issues with the training quality or course materials. Taskers may find the content unclear or difficult to understand. To address this, we should evaluate the training program to identify areas for improvement, ensuring that the material is both comprehensible and effective in preparing taskers for certification.