Dataset Link

https://www.kaggle.com/datasets/kapturovalexander/karpovs-sql-simulator

Content of the dataset

The courier service database comprises interconnected tables storing comprehensive information about the operations involved in managing food orders from users to their delivery by couriers.This database holds information crucial for the courier service operations, including details about couriers, users, their actions, and specific orders and products involved in the delivery process. The dataset facilitates tracking the lifecycle of orders, actions taken by both users and couriers, and the products being delivered, providing a comprehensive view of the courier service's functionalities and operations.

Tables and Their Contents

1. courier_actions
   A. courier_id: Integer representing the unique identifier for couriers.
   B. order_id: Integer identifying the specific order.
   C. action: Text field capturing the type of action taken by the courier (e.g., picked up, delivered).
   D. time: Text field recording the timestamp of the action.
2. couriers
   A. Courier_id : Integer serving as the unique identifier for couriers.
   B. Birth_date : Text field containing the date of birth for each courier.
   C. Sex : Text field indicating the gender of the courier.

3. products
   A. product_id: Integer uniquely identifying each product.
   B. name: Text field describing the name of the product.
   C. price: Text field containing the price information for each product.

4. user_actions
   A. User_id : Integer serving as the unique identifier for users.
   B. Order_id : Integer identifying the specific order.
   C. Action : Text field indicating the action taken by the user (e.g., placed an order, cancelled).
   D. Time : Text field recording the timestamp of the action.

5.user_orders
   A. Order_id : Integer uniquely identifying each order.
   B. Creation_time : Text field capturing the timestamp when the order was created.
   C. Product_id : Integer identifying the specific product ordered.

6.users
   A. User_id : Integer serving as the unique identifier for users.
   B. Birth_date : Text field containing the date of birth for each user.
   C. Sex : Text field indicating the gender of the user.

Business objectives

The objective of utilising the courier service database is to optimise and streamline the delivery operations while enhancing user satisfaction and efficiency. By leveraging the comprehensive dataset, the business aims to analyse various aspects of the courier service to make data-driven decisions and improvements. Specifically, the goals could include:

1. Optimising Delivery Routes and Time Management: Analysing courier_actions data alongside timestamps can help identify patterns in courier movement, allowing the business to optimise routes, reduce delivery times, and enhance overall efficiency.
2. Improving Customer Experience: Understanding user_actions and user_orders data can assist in recognizing trends in user behaviour, preferences, and order patterns. This knowledge can be utilised to personalise services, offer targeted promotions, and improve overall customer satisfaction.
3. Enhancing Courier Management: Utilising information from the couriers table and their actions, the company can improve courier allocation, training programs, and performance evaluations to ensure a high level of service.
4. Inventory and Product Management: Analysing product-related data from the products table and user_orders can help in managing inventory, predicting demand for specific products, and adjusting stock levels accordingly to avoid shortages or excess inventory.
5. Forecasting and Business Planning: Leveraging historical data from the dataset can enable predictive analytics for forecasting future demand, identifying peak times, and planning resources effectively during busy periods.
6. Cost Optimization:By analysing various aspects of the dataset such as delivery times, courier efficiency, and product popularity, the business can identify cost-saving opportunities, streamline operations, and allocate resources more effectively.

Overall, the primary business objectives would likely revolve around improving operational efficiency, enhancing customer satisfaction, and maximising profitability by leveraging insights derived from the dataset's comprehensive information on user behaviour, courier actions, and order management.

Project Insights

The objective of analysing the courier service dataset is to glean actionable insights aimed at optimising operational efficiency, improving customer service, and making data-informed decisions within the delivery logistics domain. The dataset encompasses interconnected tables detailing various aspects such as user actions, order placements, courier activities, and product information. By delving into this dataset, the courier service aims to uncover pivotal insights regarding popular products, user behaviours, order dynamics, delivery performance, and gender-specific trends. These insights will enable the refinement of delivery routes, inventory management, and targeted marketing strategies, ultimately enhancing the overall service quality and customer satisfaction.

1. What are the top 3 most ordered products?
2. On which date were the highest number of orders placed?
3. How is the distribution of order actions (placed, delivered, etc.) segmented by gender?
4. What is the count of orders handled by each courier?
5. How many orders were placed by individual users?
6. What is the average number of products per order?
7. Which product generated the highest total revenue?
8. What are the top most ordered products for male and female users, and how do their order counts differ?
9. What is the busiest day for orders (by count)?
10. How does the average delivery time vary across different couriers?

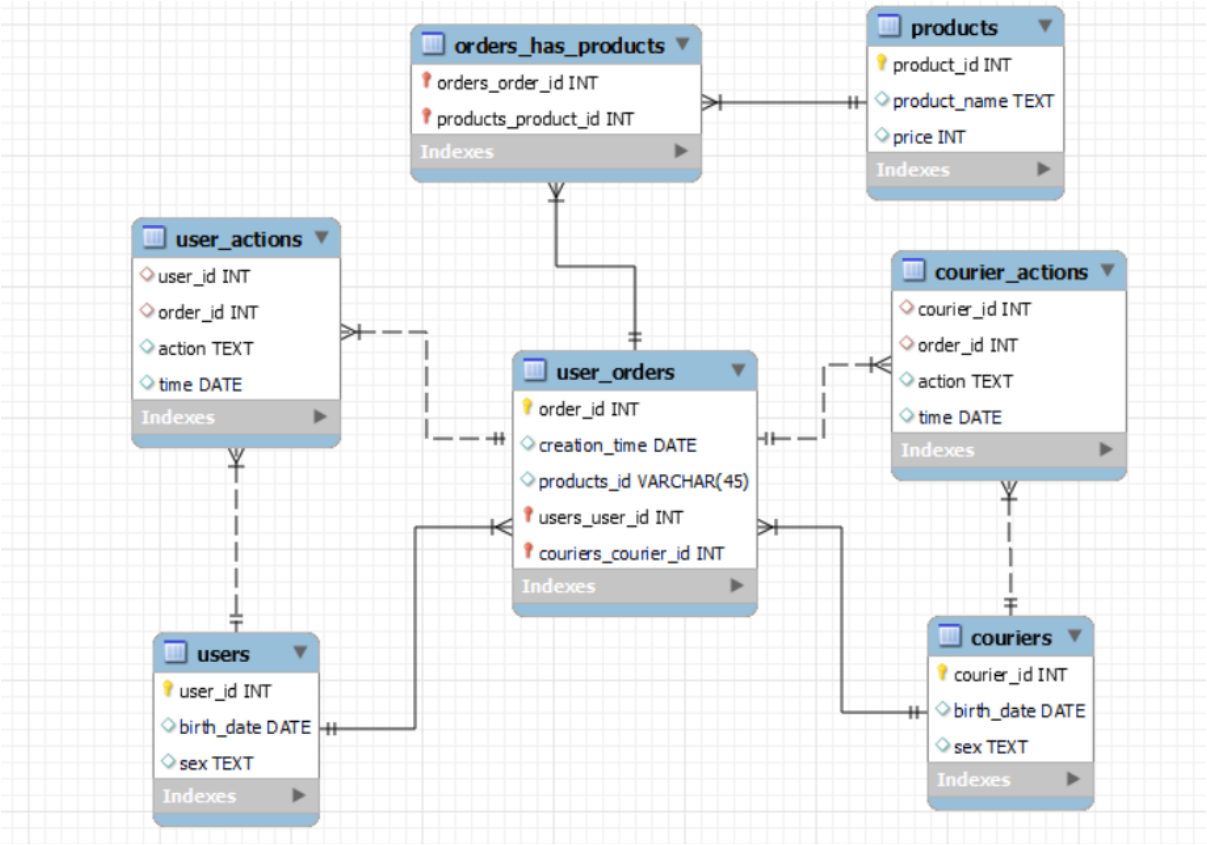
Target Audience

The primary target audience for our courier service is composed of users and couriers within the operational zones serviced by our delivery network. As a courier service operating within specific regions, our focus lies on engaging residents and businesses within these areas who regularly utilise delivery services for their orders. We target individuals who frequently place orders for food or goods and rely on timely and efficient delivery. Our audience comprises a diverse demographic, including adults aged 18 to 65, encompassing both genders, capable of utilising courier services for their delivery needs. Notably, our service caters to able-bodied individuals within the specified age range residing in the regions covered by our courier network.

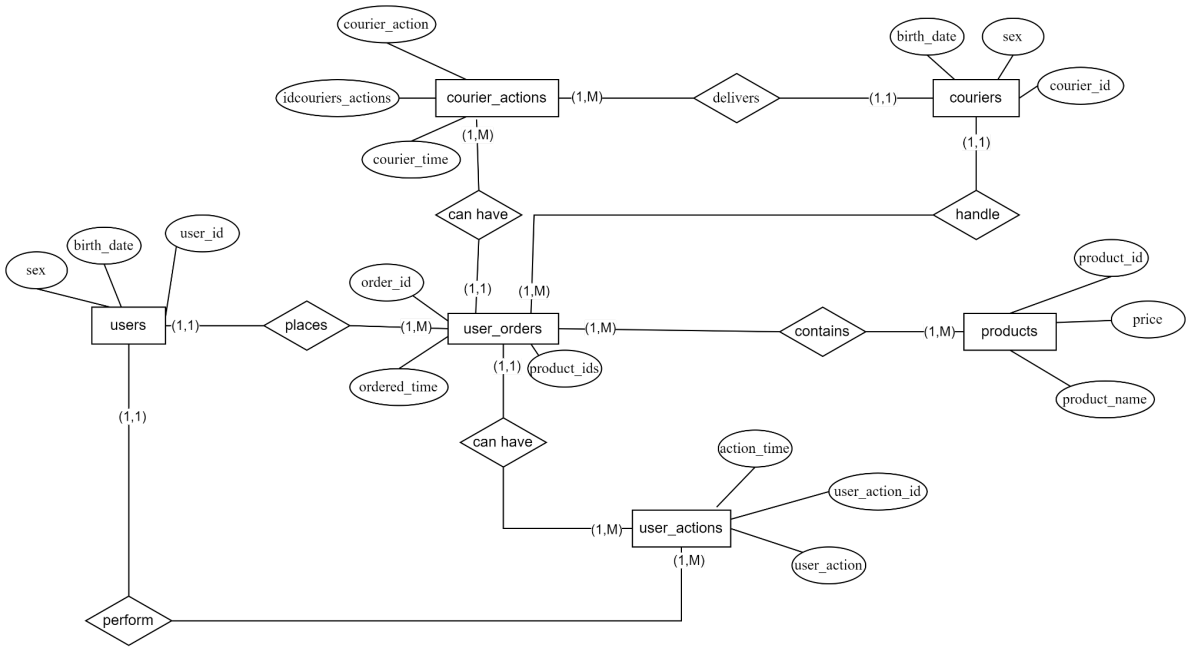
Problem Statement

The pivotal challenge faced by our courier service lies in optimising the entire delivery ecosystem by effectively managing orders, couriers, and user expectations. The core problems revolve around understanding user behaviours regarding order placement, delivery timelines, preferred products, and crucially, the logistics of the delivery process. Our dataset encompasses key information regarding order actions, user preferences, and courier activities. By leveraging this dataset effectively, we aim to address the queries revolving around order volumes, delivery times, popular products, and the relationship between user actions and courier activities. The goal is to streamline our services, ensuring prompt and efficient deliveries that cater to the dynamic needs of our user base. Ultimately, our objective is to use data-driven insights to optimise our courier service, ensuring accessibility and reliability for all users relying on our delivery network.

Logical Diagram



Conceptual Diagram



MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema deliveryservice_db
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema deliveryservice_db
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `deliveryservice_db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `deliveryservice_db` ;

-- -----------------------------------------------------
-- Table `deliveryservice_db`.`couriers`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`couriers` (
  `courier_id` INT NOT NULL,
  `birth_date` DATE NULL DEFAULT NULL,
  `sex` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`courier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `deliveryservice_db`.`users`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`users` (
  `user_id` INT NOT NULL,
  `birth_date` DATE NULL DEFAULT NULL,
  `sex` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`user_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----------------------------------------------------
-- Table `deliveryservice_db`.`user_orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`user_orders` (
  `order_id` INT NOT NULL,
  `creation_time` DATE NULL DEFAULT NULL,
  `product_id` INT NULL DEFAULT NULL,
  `users_user_id` INT NOT NULL,
  `couriers_courier_id` INT NOT NULL,
  PRIMARY KEY (`order_id`, `users_user_id`, `couriers_courier_id`),
  INDEX `fk_user_orders_users1_idx` (`users_user_id` ASC) VISIBLE,
  INDEX `fk_user_orders_couriers1_idx` (`couriers_courier_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_orders_users1`
    FOREIGN KEY (`users_user_id`)
    REFERENCES `deliveryservice_db`.`users` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_orders_couriers1`
    FOREIGN KEY (`couriers_courier_id`)
    REFERENCES `deliveryservice_db`.`couriers` (`courier_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `deliveryservice_db`.`courier_actions`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`courier_actions` (
  `courier_id` INT NULL DEFAULT NULL,
  `order_id` INT NULL DEFAULT NULL,
  `action` TEXT NULL DEFAULT NULL,
  `time` DATE NULL DEFAULT NULL,
  INDEX `fk_courier_actions_user_orders1_idx` (`order_id` ASC) VISIBLE,
  INDEX `fk_courier_actions_couriers1_idx` (`courier_id` ASC) VISIBLE,
  CONSTRAINT `fk_courier_actions_couriers1`
    FOREIGN KEY (`courier_id`)
    REFERENCES `deliveryservice_db`.`couriers` (`courier_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_courier_actions_user_orders1`
    FOREIGN KEY (`order_id`)
    REFERENCES `deliveryservice_db`.`user_orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

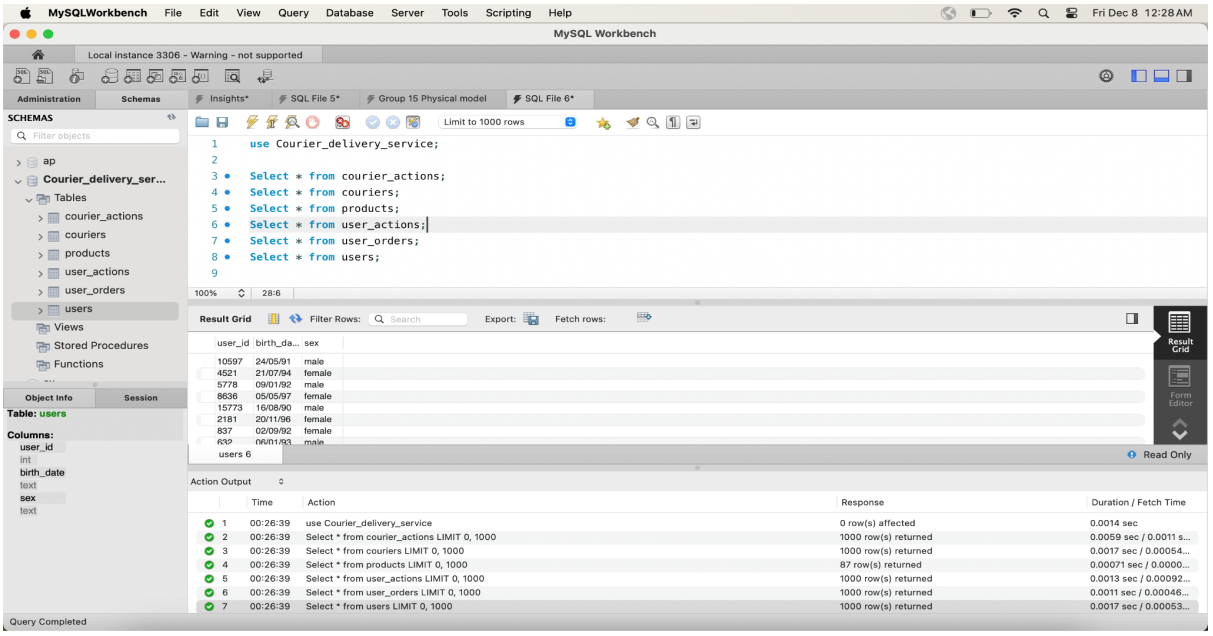
```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `deliveryservice_db`.`products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`products` (
  `product_id` INT NOT NULL,
  `product_name` VARCHAR(255) NULL DEFAULT NULL,
  `price` INT NULL DEFAULT NULL,
  PRIMARY KEY (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `deliveryservice_db`.`user_actions`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `deliveryservice_db`.`user_actions` (
  `user_id` INT NULL,
  `order_id` INT NULL DEFAULT NULL,
  `action` TEXT NULL DEFAULT NULL,
  `time` DATE NULL DEFAULT NULL,
  INDEX `fk_user_actions_user_orders1_idx` (`order_id` ASC) VISIBLE,
  INDEX `fk_user_actions_users_idx` (`user_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_actions_users`
    FOREIGN KEY (`user_id`)
    REFERENCES `deliveryservice_db`.`users` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_actions_user_orders1`
    FOREIGN KEY (`order_id`)
    REFERENCES `deliveryservice_db`.`user_orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Final Outcome



1.What are the top 3 most ordered products?

```
10
11      /* 1. Top 3 most ordered products: */
12      SELECT p.product_id,p.name,COUNT(uo.product_id) AS order_count
13      FROM user_orders uo
14      JOIN products p ON uo.product_id = p.product_id
15      GROUP BY p.product_id, p.name
16      ORDER BY order_count DESC
17      limit 3;
18
```

100% 9:17

Result Grid Filter Rows: Search Export: Fetch rows:

product_id	name	order_count
40	bread	4161
77	chicken	4144
1	sugar	4130

Final Analysis : The top three most frequently ordered products, namely Bread, Chicken, and Sugar, indicate significant customer interest and demand. This insight suggests these products are popular among customers and might be key revenue drivers. Analysing these products further for their profit margins, trends in demand over time, or association with other products could help in devising targeted marketing strategies, ensuring adequate stock availability, and potentially introducing product bundles or promotions to leverage their popularity.

2. On which date were the highest number of orders placed?

```
18
19    /* 2. Date at which highest number of orders has been placed? */
20    SELECT COUNT(order_id) AS max_count, cast(ca.time AS date ) AS delivery_date
21       FROM courier_actions ca group by delivery_date order by max_count DESC;
22
23
```

100% 26:16

Result Grid Filter Rows: Search Export:

max_count	delivery_date
12296	2004-09-22
11714	2003-09-22
11277	2008-09-22
10240	2002-09-22
9788	2007-09-22
9661	2005-09-22
7574	2001-09-22
7078	2006-09-22
7011	2031-08-22
6953	2030-08-22
6721	2029-08-22
6118	2028-08-22
4377	2027-08-22
2971	2026-08-22
2154	2025-08-22
278	2024-08-22

Final Analysis : Identifying the busiest day for order placements, 2004-09-22, provides critical information about customer behaviour and demand patterns. Understanding the factors contributing to this surge in orders, whether they're related to seasonal trends, promotional activities, or external events, enables businesses to capitalise on such occurrences in the future. It facilitates better resource allocation, staffing adjustments, and promotional planning to optimise sales during similar periods.

3. How is the distribution of order actions (placed, delivered, etc.) segmented by gender?

```
22
23    /* 3. What is the distribution of order actions (placed, delivered, etc.) by gender? */
24    SELECT u.sex, ua.action, COUNT(ua.order_id) AS action_count
25    FROM users u
26    JOIN user_actions ua ON u.user_id = ua.user_id
27    GROUP BY u.sex, ua.action;
28
```

100% 75:21

Result Grid Filter Rows: Search Export:

sex	action	action_count
female	create_order	28146
female	cancel_order	1434
male	create_order	28362
male	cancel_order	1404

Final Analysis : Observing the distribution of order actions between genders highlights insights into user behaviour. For instance, the data indicates that males tend to place more orders compared to females. Understanding these disparities in user engagement could help tailor marketing campaigns, product offerings, or user experience improvements to cater to the preferences and behaviours of different gender demographics, potentially increasing engagement and customer satisfaction.

4. What is the count of orders handled by each courier?

```
28
29      /*  4. What are the number of orders per courier? */
30 •    SELECT courier_id, COUNT(*) AS num_orders_handled
31      FROM courier_actions
32      GROUP BY courier_id
33      ORDER BY num_orders_handled DESC;
34
```

100% 47:26

Result Grid | Filter Rows: Search | Export: | Fetch rows:

courier_id	num_orders_handled
252	108
291	108
492	108
708	107
23	105
276	103
501	103
329	101
203	100
66	100
206	100
179	99
40	99

Result 12

Final Analysis : Determining the number of orders handled by each courier provides insights into their workload and efficiency. This information aids in assessing courier performance and workload distribution. By identifying the busiest couriers or those managing a higher volume of orders, service providers can implement strategies to optimise routes, redistribute workload, or provide additional support where necessary, ensuring timely and efficient deliveries.

5. How many orders were placed by individual users?

```
34
35      /* 5. what is the number of orders placed by an user ? */
36      SELECT  user_id,   action,   COUNT(action) AS action_count
37      FROM  user_actions
38      GROUP BY  user_id, action
39      ORDER BY  action_count DESC;
40
```

100% 20:32

Result Grid | Filter Rows: Search | Export: | Fetch rows:

user_id	action	action_count
3793	create_order	18
3922	create_order	15
362	create_order	15
3333	create_order	14
803	create_order	14
3941	create_order	14
12	create_order	13
4046	create_order	13
1170	create_order	13
4015	create_order	13
763	create_order	13
1533	create_order	13
2077	create_order	13

Final Analysis : Identifying users with the highest number of orders, such as those with IDs 3793, 3922, 362 sheds light on highly engaged customers. Understanding their preferences, purchase behaviours, or loyalty patterns could be instrumental in implementing loyalty programs, personalised marketing strategies, or targeted promotions aimed at retaining these valuable customers and encouraging continued engagement.

6. What is the average number of products per order?

```
41      /* 6. Average number of products per order: */
42      SELECT AVG(num_products) AS avg_products_per_order
43      FROM (
44          SELECT order_id, COUNT(*) AS num_products
45          FROM user_orders
46          GROUP BY order_id
47      ) AS product_counts;
48
```

100% 26:38

Result Grid | Filter Rows: Search | Export:

avg_products_per_or...
3.3978

Final Analysis : The average number of products per order, calculated as 3, serves as a crucial metric for inventory management and understanding customer buying habits. Analysing this metric over time or in conjunction with specific product categories can assist in optimising inventory levels, predicting demand fluctuations, or strategically bundling products to encourage higher order values. Moreover, it

helps in refining marketing strategies to promote complementary products or upsell strategies based on typical purchase patterns.

7. Which product generated the highest total revenue?

```
48
49      /* 7. Most popular product by total revenue generated: */
50      SELECT p.name AS popular_product, SUM(p.price) AS total_revenue
51      FROM user_orders uo
52      JOIN Products p ON uo.product_id = p.product_id
53      GROUP BY p.name
54      ORDER BY total_revenue DESC
55      LIMIT 1;
56
57
```

100% 21:45

Result Grid | Filter Rows: Search | Export: | Fetch rows:

popular_product	total_revenue
pork	1437750

Final Analysis : Identifying the product generating the highest total revenue, such as Pork, signifies its substantial contribution to overall sales. Understanding the factors driving its popularity, such as unique features, competitive pricing, or customer preference, allows businesses to capitalise on its success. It opens opportunities for targeted marketing campaigns, cross-selling or upselling strategies, or product diversification efforts aimed at leveraging this product's success to maximise revenue generation.

8. What are the top most ordered products for male and female users, and how do their order counts differ?

```sql
/* 8. What are the top most ordered products for both male and female users, and how do they differ in terms of order count? */
WITH RankedProducts AS (
    SELECT
        u.sex,
        p.name,
        COUNT(o.product_id) AS order_count,
        ROW_NUMBER() OVER (PARTITION BY u.sex ORDER BY COUNT(o.product_id) DESC) AS product_rank
    FROM users u
    JOIN user_actions ua ON u.user_id = ua.user_id
    JOIN user_orders o ON ua.order_id = o.order_id
    JOIN products p ON o.product_id = p.product_id
    WHERE ua.action = 'create_order'
    GROUP BY u.sex, p.name
)
SELECT
    sex,name AS most_ordered_product,order_count
FROM RankedProducts
WHERE product_rank = 1;
```

100% 16:53

Result Grid Filter Rows: Search Export:

sex	most_ordered_prod...	order_count
female	bread	2002
male	bananas	1975

Final Analysis : Differentiating the most ordered products for male and female users, such as bananas for males and bread for females, reveals gender-specific preferences. Understanding these preferences helps in tailoring marketing initiatives, product assortments, or promotional offers to cater to the distinct tastes or interests of each gender demographic. By addressing these preferences, businesses can enhance customer satisfaction, boost engagement, and potentially expand their customer base within specific demographic segments.

9. What is the busiest day for orders (by count)?

```sql
/* 9. Busiest day for orders (by count): */
SELECT creation_time AS order_date, COUNT(*) AS total_orders
FROM user_Orders
GROUP BY order_date
ORDER BY total_orders DESC
LIMIT 1;
```

100% 20:73

Result Grid Filter Rows: Search Export: Fetch rows:

order_date	total_orders
2/9/2022 20:40	80

Final Analysis : Identifying 02/09/2022 as the busiest day for order placements provides crucial insights into peak customer demand periods. This information allows businesses to anticipate and prepare for such peaks in the future. Whether it's ensuring sufficient inventory levels, staffing adjustments, or targeted promotions during these high-demand periods, being prepared helps in maximising sales opportunities, enhancing customer service, and maintaining operational efficiency.

10. How does the average delivery time vary across different couriers?

```
82
83    /*10. How does the average delivery time vary across different couriers? */
84    SELECT courier_id,AVG(CAST(time AS SIGNED)) AS average_delivery_time
85    FROM courier_actions
86    WHERE action = 'deliver_order'
87    GROUP BY courier_id
88    ORDER BY average_delivery_time DESC;
89
90
```

100% 44:76

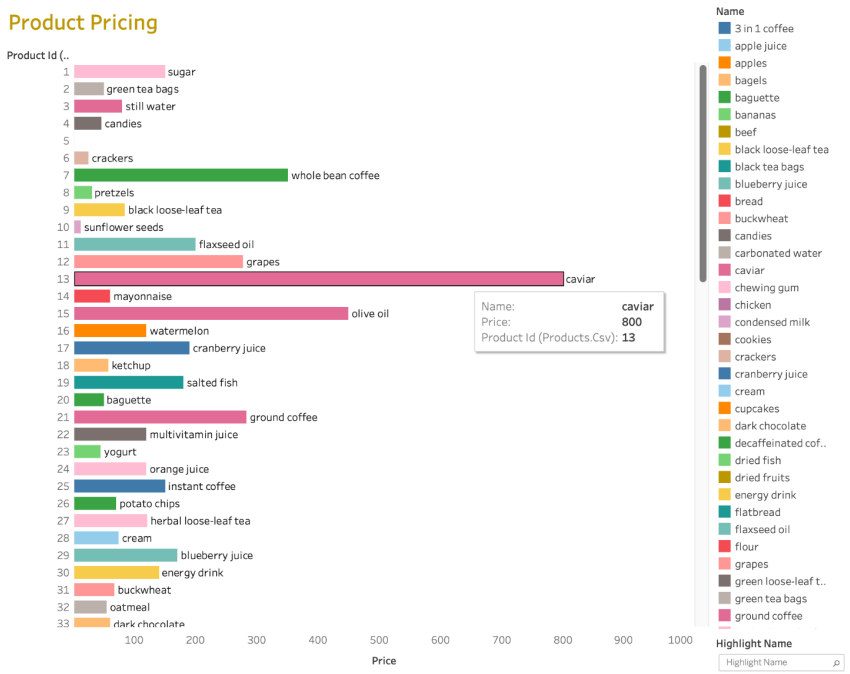
Result Grid Filter Rows: Search Export: Fetch rows:

courier_id	average_delivery_ti...
544	23.5000
328	22.8276
214	22.4848
490	21.8333
572	21.6250
545	21.6250
4	21.6129
251	21.5882
736	21.4483
103	21.3023
179	21.2653
770	21.0385
642	20.9024
81	20.8919
13	20.8537

Final Analysis : Analysing the average delivery time for different couriers, particularly noting the shortest average delivery time for Courier_ID : 544 , highlights variations in delivery efficiency. Understanding the factors contributing to these variances, whether it's route optimization, delivery protocols, or individual courier performance, enables service providers to implement best practices across their delivery network. It ensures consistent service standards, improved customer satisfaction, and potential operational enhancements to streamline delivery processes.

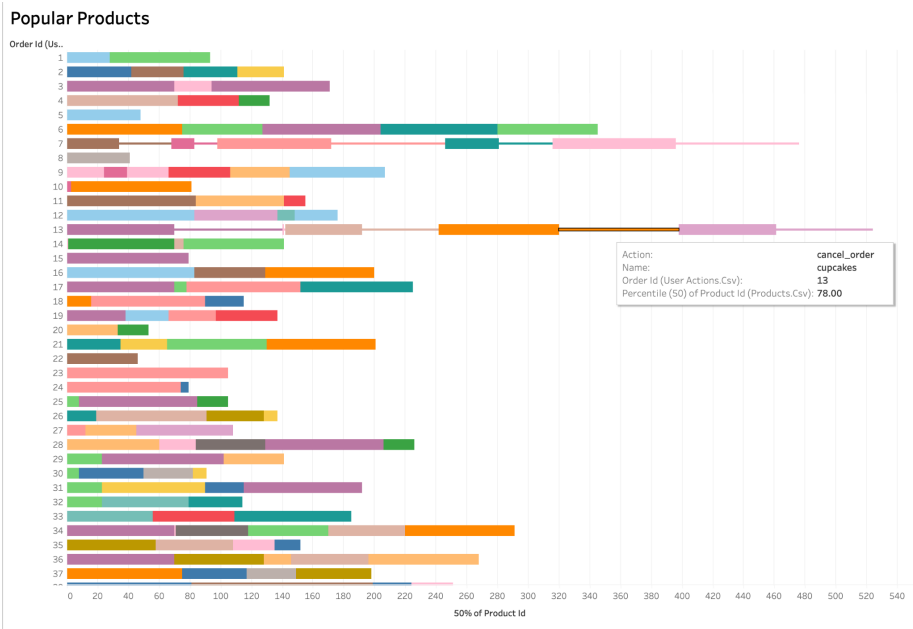
Visualisation charts:

1. What is the top-priced product in the dataset?



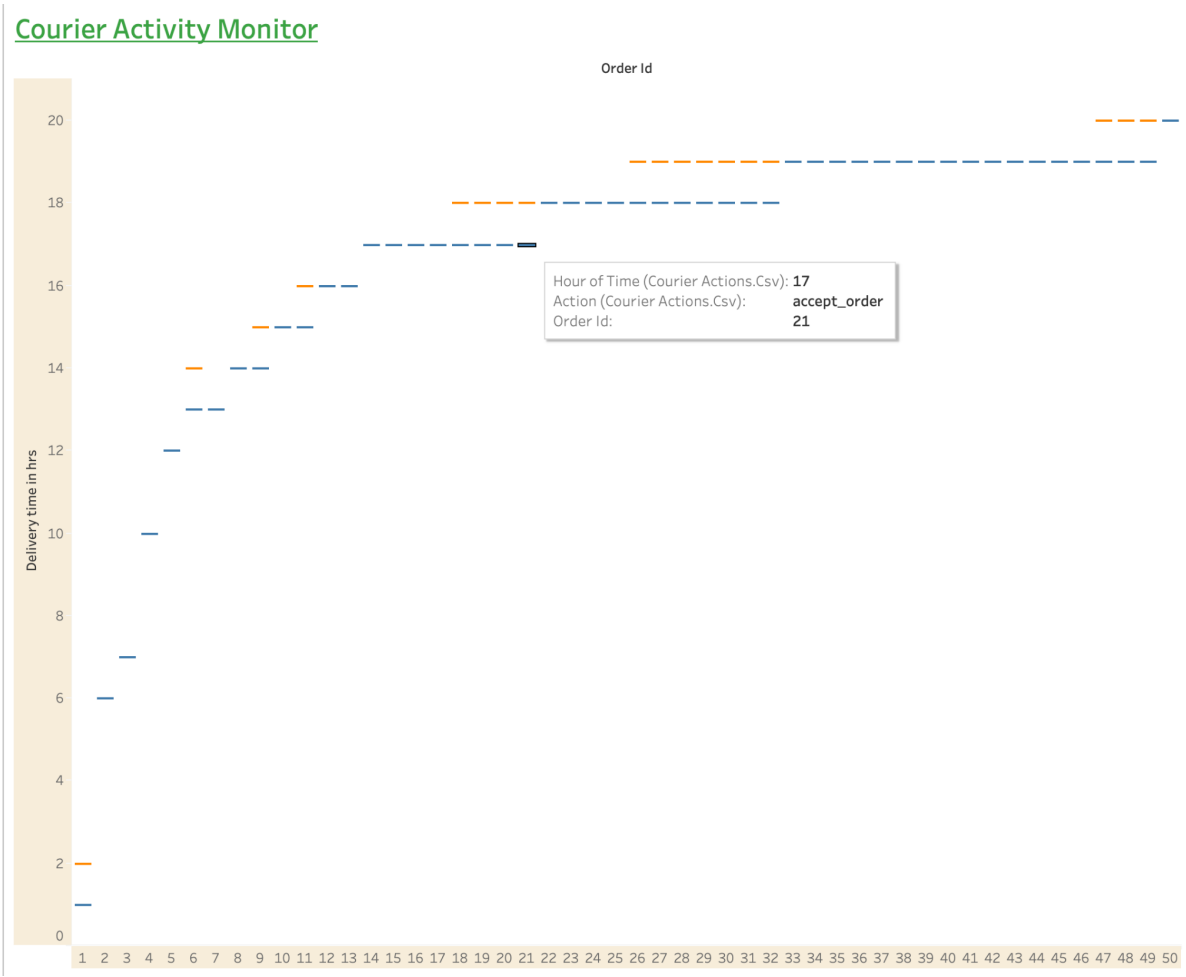
By visualising the product prices, it's apparent that the highest-priced product is caviar, valued at 800. This insight highlights the premium product in the inventory.

2. What products are frequently cancelled and created among the first 50 orders?



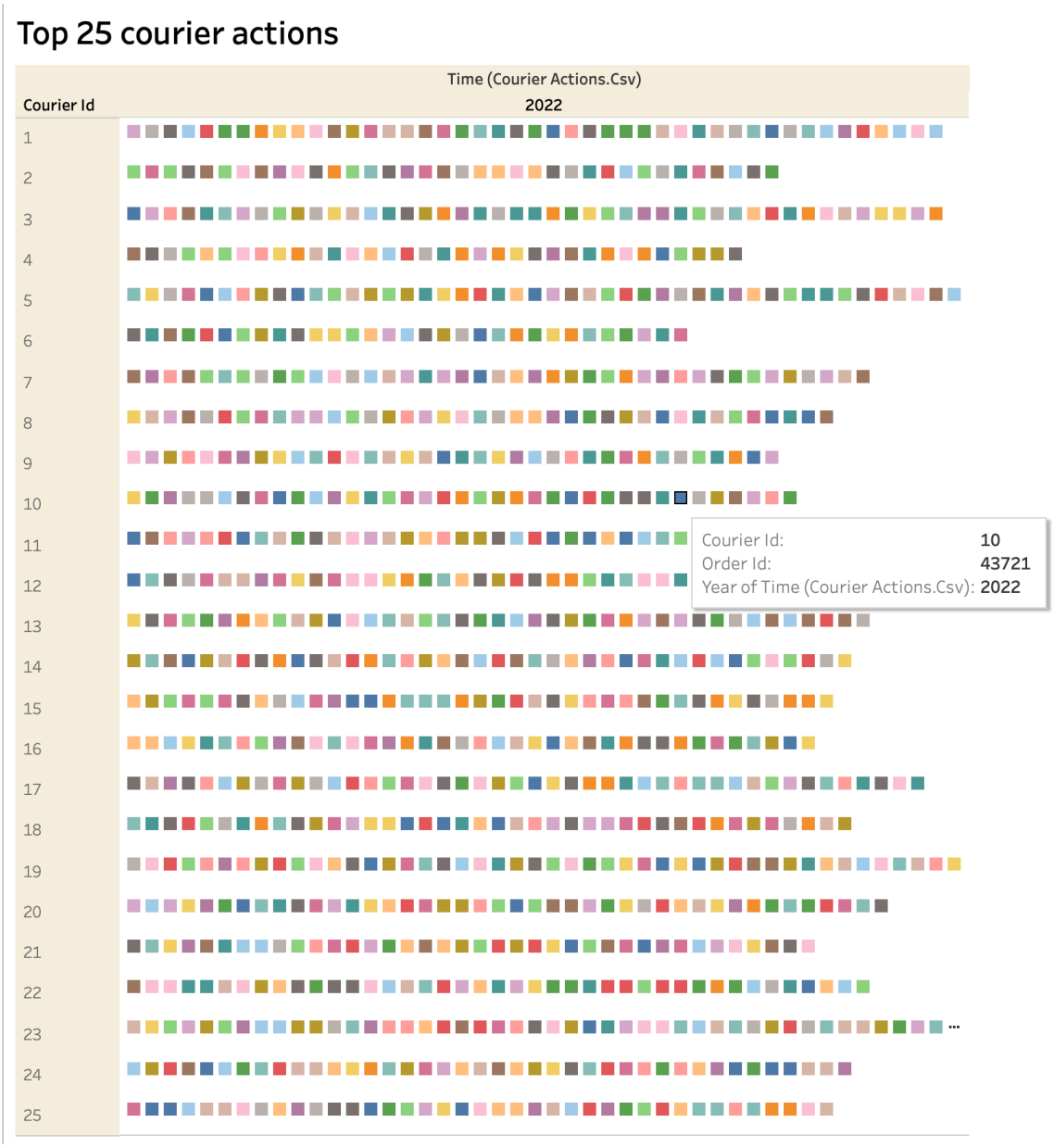
The visualisation reveals specific products that are commonly cancelled and created, with cupcakes emerging as the most frequently alternated product. This insight aids in understanding customer preferences and potential issues with specific items.

3. What are the courier actions undertaken by the top 25 couriers, including order IDs and action times?



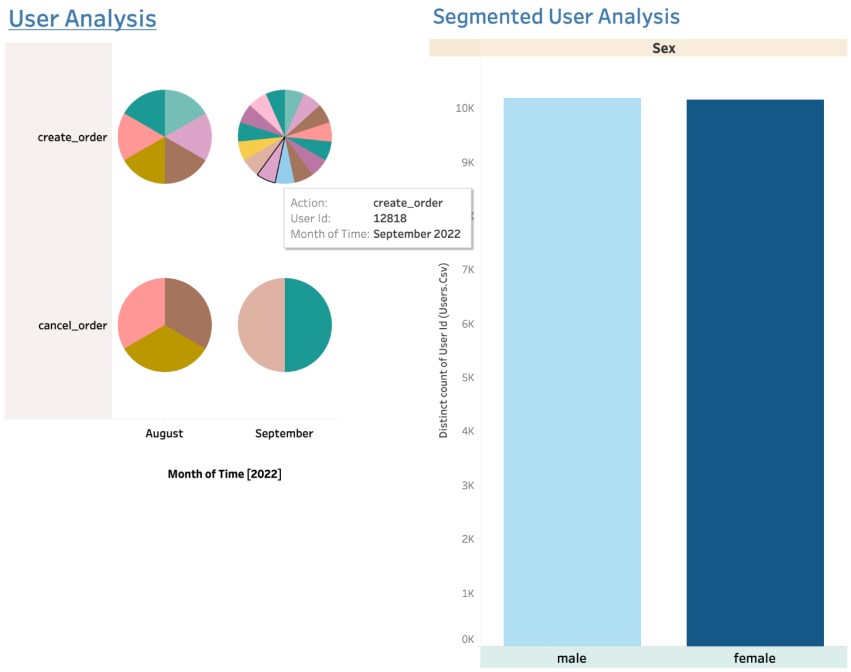
Through the displayed graph, it illustrates the actions performed by the top 25 couriers along with their respective action times. This analysis provides a glimpse into courier performance and their responsiveness in executing actions.

4. What is the estimated time taken by couriers to accept and deliver an order?



The displayed Gantt chart indicates a general estimated courier time of 1 hour for order acceptance and progression towards delivery. This visualisation offers insights into the typical courier order processing timeframe, aiding in assessing efficiency and service standards.

5. What is the median age of users, segmented by gender?



The visual representation indicates a consistent median age among users across genders. This analysis provides an overview of user demographics based on age, essential for understanding the user base and tailoring services accordingly.

Conclusion

Throughout our analysis leveraging SQL queries on multiple datasets within the courier service, we've uncovered pivotal insights crucial for enhancing operational efficiency and customer-centric strategies. These insights span various aspects that are integral to the success of our courier service.

Our queries have revealed patterns such as top-selling products, gender-specific preferences, peak order periods, and courier efficiency. Understanding these patterns provides actionable information for inventory management, targeted marketing campaigns, and service optimization. For instance, recognizing the most ordered products allows us to strategically manage our stock, while identifying peak order days aids in resource allocation and service preparedness.

Moreover, insights into customer behaviour, including the preferences of high-ordering users and average products per order, offer pathways for enhancing customer experiences. Tailoring services to cater to these preferences and habits could significantly improve customer satisfaction and retention rates.

By leveraging these insights, we can implement focused strategies to enhance

operational efficiency, improve customer engagement, and streamline our delivery services. These efforts can lead to an optimised workflow, better resource utilisation, and ultimately an elevated service standard, ensuring customer satisfaction and potentially fostering business growth.

In conclusion, the SQL-driven analyses have provided valuable insights that act as a roadmap for us to make informed decisions, optimise our operations, and offer enhanced services that align more closely with customer needs. This approach positions us for sustained success and growth within the industry.