# Database Management (CSC 410)


1. Olorunfemi-Ojo Daniel Tomiwa 190805503
2. Sanusi Adeyemi Abdulazeez 180805050
3. Muhammed Sodiq Akande 180805056
4. Ezeani David 180805081
5. Ogunrinde Motunrayo Deborah 190805514

## Table of Contents

# 1. Introduction

## 1.1. Statement of Problem

Model and design a database for Taaskly Service (https://taaskly.xyz).

## 1.2. Organisation Overview

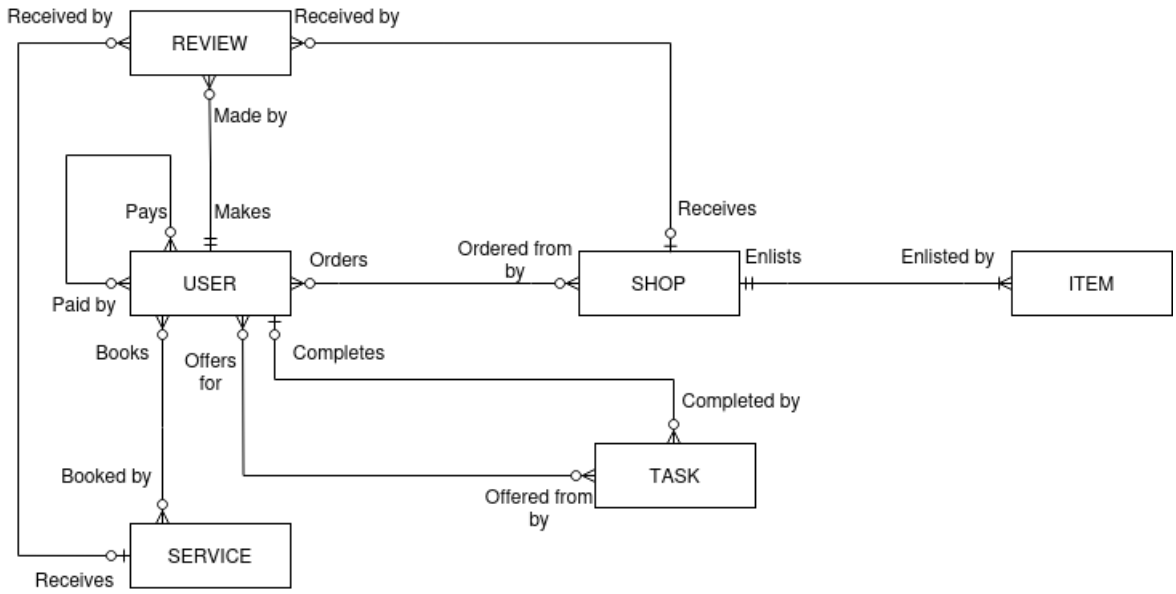
Taaskly is a platform that allows individuals to create listings for tasks they need to have done, as well as products and services they offer. Users who require assistance can post their tasks, while those who offer products and services can create a listing to increase their visibility. Customers can then browse the listings, read reviews, and use our secure payment feature to ensure that their funds are held safely until the business deal is completed. Users use the payment feature to pay for services or products offered through on Taaskly. Funds are held until the service is completed and released to the service provider only after the customer confirms the completion of the service.

# 2. Analysis

## 2.1. Entities

- USER: a customer on the Taaskly platform
- SHOP: a vendor selling products
- ITEM: a product enlisted by a vendor
- SERVICE: an activity rendered
- TASK: an activity to be undertaken
- REVIEW: a comment about an experience

## 2.2. Enterprise ER Diagram




## 2.3. Business Rules

- A USER may make many orders from many SHOPs. Each SHOP may be ordered from by many USERs.
- A USER may book many SERVICEs. Each SERVICE may be booked by many USERs.
- A USER may complete many TASKs. Each TASK may be completed by a USER.
- A USER may offer to complete many TASKs. Each TASK may receive offers by many USERs.
- A USER may pay other USERs. Each USER may be paid by other USERs.
- A SHOP must enlist one or more ITEMs. Each ITEM must be enlisted by a SHOP.
- A USER may make many REVIEWs. Each REVIEW must be made by a USER.

- A SHOP may receive many REVIEWs. Each REVIEW may be received by a SHOP.
- A SERVICE may receive many REVIEWs. Each REVIEW may be received by a SERVICE.

## 2.4. Entity Modelling

### USER

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| User_Id(PK) | int | User's identifier | 4112 |
| User_Email | char(50) | User's email | p45@gmail.com |
| User_Nickname | char(25) | Username | user_p45 |
| User_Firstname | char(25) | User's firstname | Blossom |
| User_Lastname | char(25) | User's lastname | Dmitri |
| User_Bio | varchar(1000) (optional) | User's biography | I am a genius, the myth and the legend. |
| User_Phone | char(13) | User's phone no. | 08012345678 |
| User_Balance | double | User's wallet balance | 420.99 |
| User_Referral_Code | char(6) | User's referral code | TSKP45 |
| User_Profile_Picture | blob (optional) | User's profile picture | |

### SHOP

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Shop_Id(PK) | int | Shop's identifier | 231 |
| User_Id(FK) | int | Shop's owner | 3143 |
| Shop_Name | char(50) | Shop's name | The Yam That I Yam Stores |
| Shop_Profile_Picture | blob (optional) | Shop's profile picture | |
| Shop_Bio | varchar(1000) | Shop's description | We sell food items |
| Shop_Phone | char(13) | Shop's phone no. | 08012345678 |
| (Shop_Location) | | (Street, City, State) | (Olise St, Yaba, Lagos) |

### ITEM

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Item_Id(PK) | int | Item identifier | 419 |

| | | | |
|---|---|---|---|
| Shop_Id(FK) | int | Item seller | 321 |
| Item_Name | char(50) | Item name | Azkaban Loud |
| Item_Description | char(1000) | Item description | Hogwarts premium enchanted cigarettes |
| Item_Price | double | Item price | 4323.00 |
| Item_Image | blob | Item picture | |
| {Item_Category} | | Item category | {Food, Grocery} |

# SERVICE

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Service_Id(PK) | int | Service identifier | 3219 |
| User_Id(FK) | int | Service owner | 321 |
| Service_Name | char(50) | Service name | Otunba Wire Wire |
| Service_Bio | char(1000) | Service description | Electrical fittings and wiring |
| Service_Rating | double | Service rating | 3.5 |
| Service_Phone | char(13) | Service phone no. | 08012345678 |
| Service_Profile_Picture | blob | Service profile picture | |
| (Service_Location) | | (Street, City, State) | (Olise St, Yaba, Lagos) |
| {Service_Category} | | Service category | {Electronics, Electrical} |

# TASK

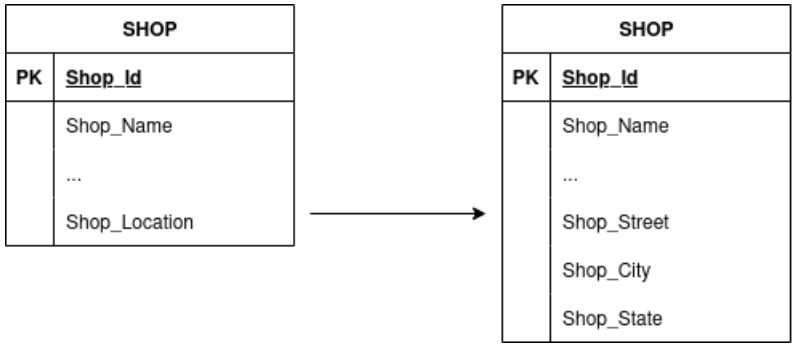
| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Task_Id(PK) | int | Task's identifier | 8983 |
| User_Id(FK) | int | Task's creator | 6782 |
| Task_Description | char(1000) | Task's description | I need a french language tutor |
| Task_Price | double | Amount to be paid | 4257.99 |
| Task_Status | char(10) | Task's status | Opened/Closed |
| Task_Executor(FK) | int | Task's executor | 323 |
| Task_Expiration_Date | datetime | Application deadline | 23/03/2015 03:30:00 |
| {Task_Category} | | Task's category | {Linguistics, Teaching} |
| (Task_Offers) | | (Description, Price) | (Je ne sais quois, va va voom!!, 50000.00) |

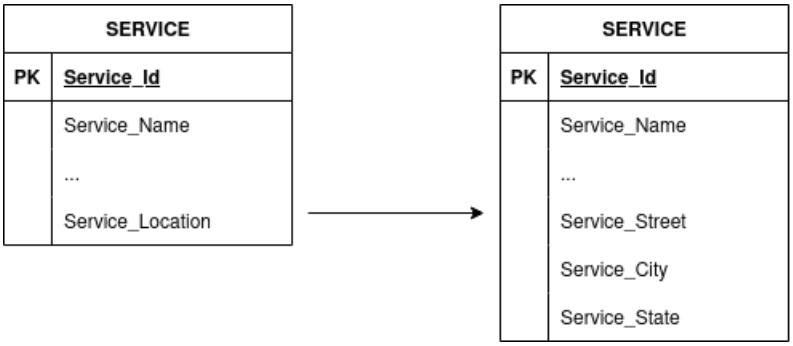
# REVIEW

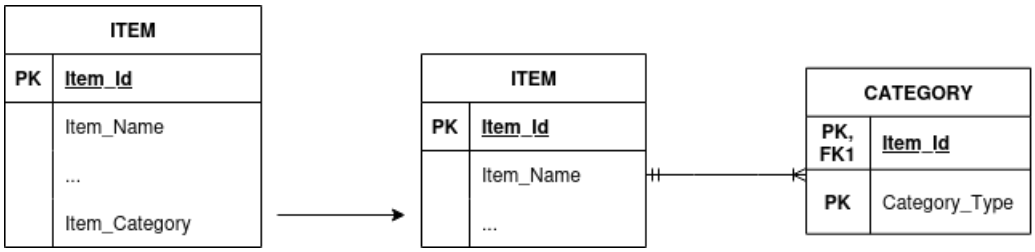
| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Review_Id(PK) | int | Review's identifier | 8983 |
| User_Id(FK) | int | Review creator | 6782 |
| Review_Comment | char(1000) | Review left by user | Very very poor. Caused an electrical fire in my home |
| Review_Rating | double | User rating | 3.5 |
| Review_Timestamp | datetime | Review date and time | 23/04/2033 09:33:02AM |
| Shop_IdFK) | int | Shop's identifier | 6533 |
| Service_Id(FK) | int | Service identifier | 345544 |

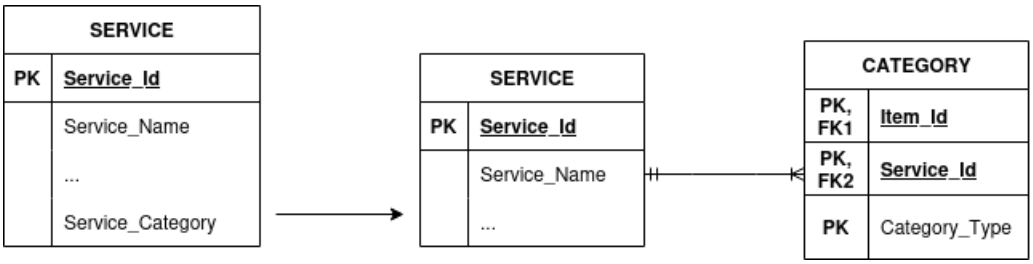
# 3. Design

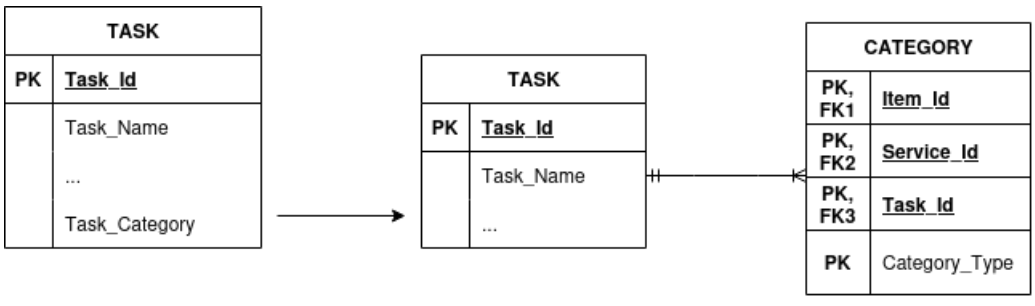
## 3.1. Mapping ER Diagram to Relations

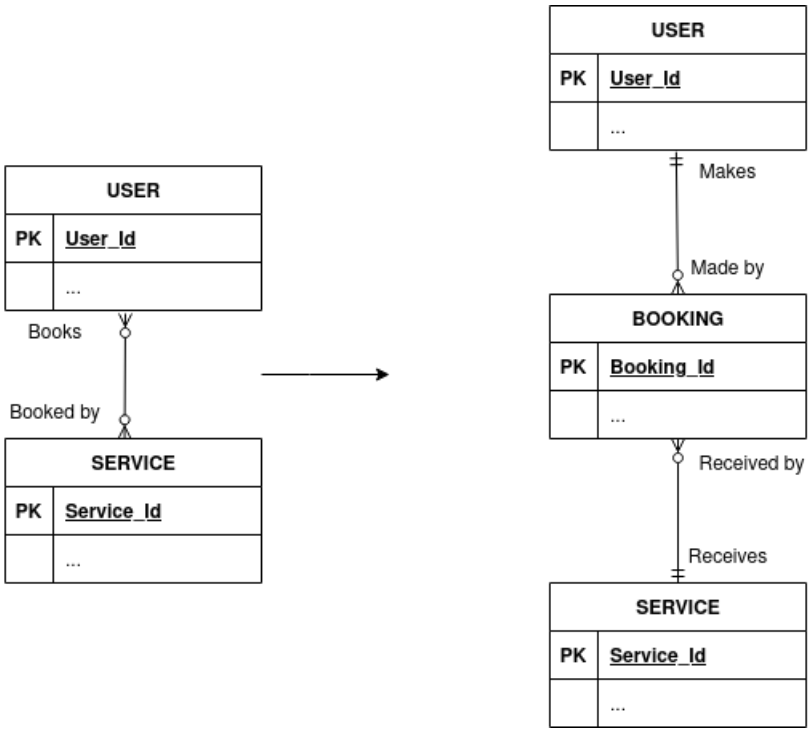


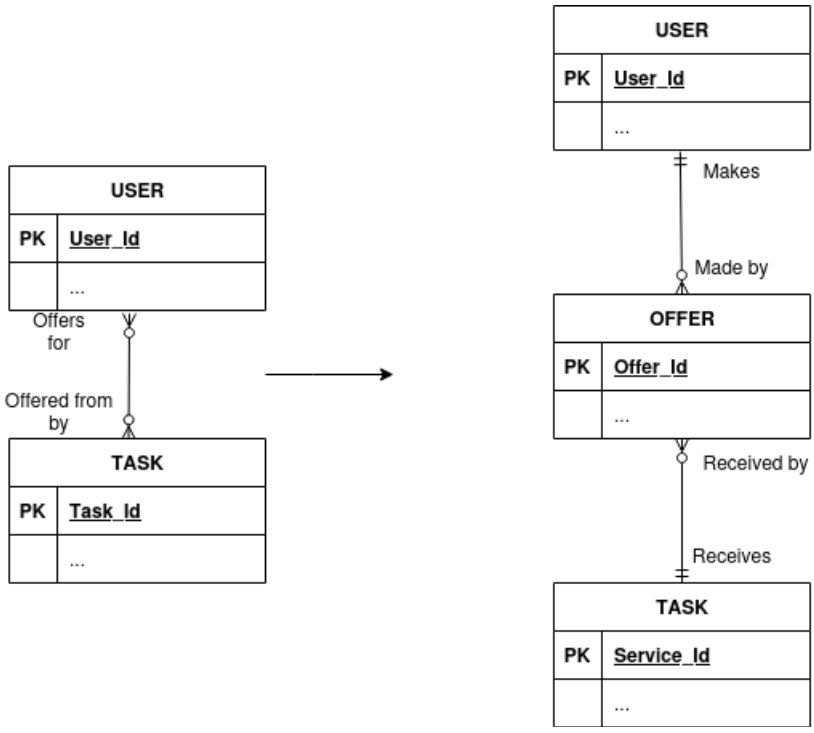



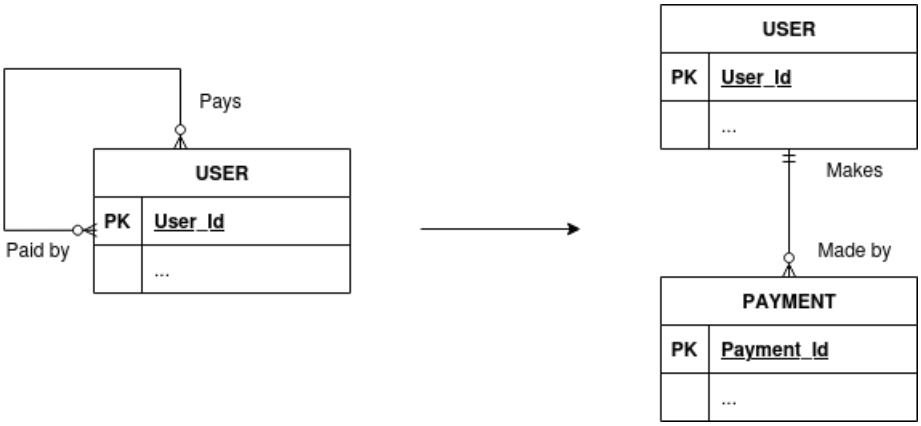



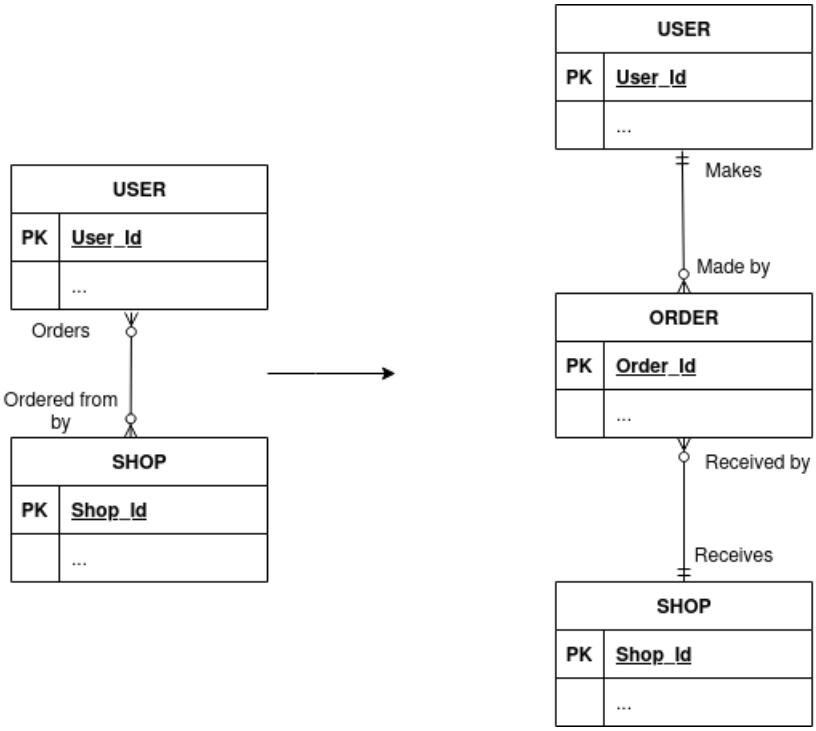



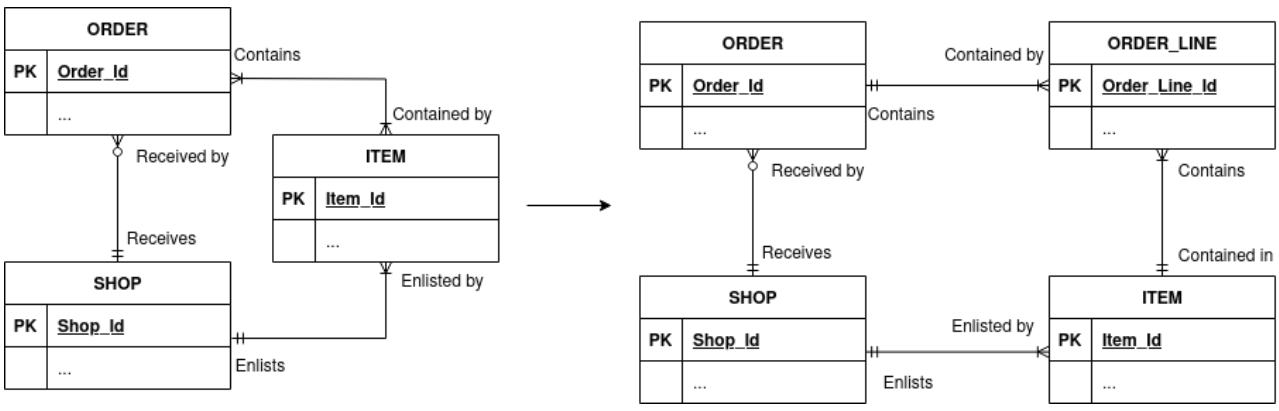



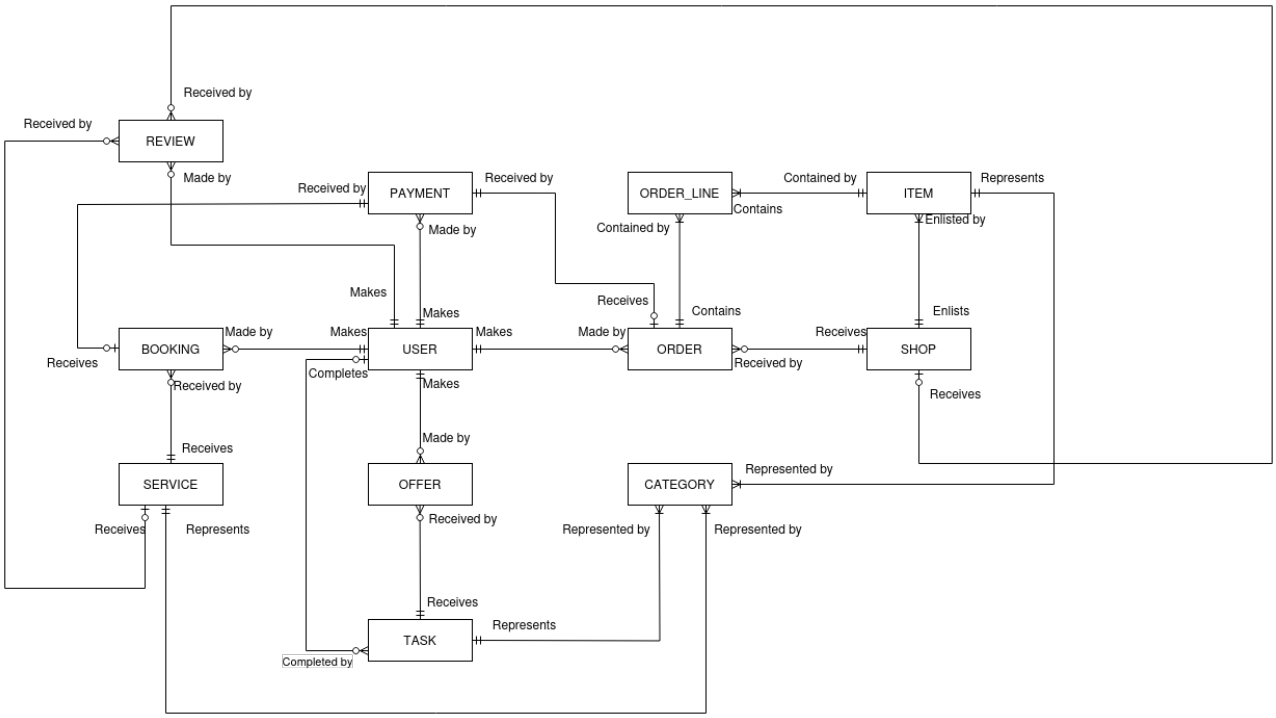

USER
PK
User_Id
...
Books
Booked by
SERVICE
PK
Service_Id
...
USER
PK
User_Id
...
Makes
Made by
BOOKING
PK
Booking_Id
...
Received by
Receives
SERVICE
PK
Service_Id
...


USER
PK
User_Id
...
Offers for
Offered from by
TASK
PK
Task_Id
...
USER
PK
User_Id
...
Makes
Made by
OFFER
PK
Offer_Id
...
Received by
Receives
TASK
PK
Service_Id
...


Pays
USER
PK
User_Id
...
Paid by
USER
PK
User_Id
...
Makes
Made by
PAYMENT
PK
Payment_Id
...


USER
PK
User_Id
...
Orders
Ordered from by
SHOP
PK
Shop_Id
...
USER
PK
User_Id
...
Makes
Made by
ORDER
PK
Order_Id
...
Received by
Receives
SHOP
PK
Shop_Id
...


ORDER
PK
Order_Id
...
Contains
Contained by
Received by
ITEM
PK
Item_Id
...
Receives
Enlisted by
SHOP
PK
Shop_Id
...
Enlists
ORDER
PK
Order_Id
...
Contained by
Contains
ORDER_LINE
PK
Order_Line_Id
...
Received by
Contains
Receives
Contained in
SHOP
PK
Shop_Id
...
Enlisted by
Enlists
ITEM
PK
Item_Id
...




# USER

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| User_Id(PK) | int | User’s identifier | 4112 |
| User_Email | char(50) | User’s email | p45@gmail.com |
| User_Nickname | char(25) | Username | user_p45 |
| User_Firstname | char(25) | User’s firstname | Blossom |
| User_Lastname | char(25) | User’s lastname | Dmitri |
| User_Bio | varchar(1000) (optional) | User’s biography | I am a genius, the myth and the legend. |
| User_Phone | char(13) | User’s phone no. | 08012345678 |
| User_Balance | double | User’s wallet balance | 420.99 |
| User_Referral_Code | char(6) | User’s referral code | TSKP45 |
| User_Profile_Picture | blob (optional) | User’s profile picture | |

# SHOP

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Shop_Id(PK) | int | Shop’s identifier | 231 |
| Shop_Name | char(50) | Shop’s name | The Yam That I Yam Stores |
| Shop_Profile_Pictur | blob | Shop’s profile | |

| | | | |
|---|---|---|---|
| e | (optional) | picture | |
| Shop_Bio | varchar(1000) | Shop’s description | We sell food items |
| Shop_Phone | char(13) | Shop’s phone no. | 08012345678 |
| Shop_Street | char(25) | Shop’s street | Olise St |
| Shop_City | char(25) | Shop’s city | Yaba |
| Shop_State | char(25) | Shop’s state | Lagos |

## ITEM

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Item_Id(PK) | int | Item identifier | 419 |
| Shop_Id(FK) | int | Item seller | 321 |
| Item_Name | char(50) | Item name | Azkaban Loud |
| Item_Description | char(1000) | Item description | Hogwarts premium enchanted cigarettes |
| Item_Price | double | Item price | 4323.00 |
| Item_Image | blob | Item picture | |

## ORDER

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Order_Id(PK) | int | Order identifier | 419 |
| Shop_Id(FK) | int | Vendor’s identifier | 321 |
| User_Id(FK) | int | Identifier of user making order | 3 |
| Order_Cost | double | Order’s cost | 4232.00 |

## ORDER_LINE

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Order_Line_Id(PK) | int | Order line’s identifier | 419 |
| Order_Id(FK) | int | Order’s identifier | 321 |
| Item_Id(FK) | int | Item’s identifier | 430 |
| Order_Line_Item_Quantity | int | Item’s quantity | 2 |
| Order_Line_Cost | double | Order line’s cost | 333.85 |

## SERVICE

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Service_Id(PK) | int | Service identifier | 3219 |
| Service_Name | char(50) | Service name | Otunba Wire Wire |
| Service_Bio | varchar(1000) | Service description | Electrical fittings and wiring |
| Service_Phone | char(13) | Service phone no. | 08012345678 |
| Service_Profile_Picture | blob | Service profile picture | |
| Service_Street | char(25) | Service street | Olise St |
| Service_City | char(25) | Service city | Yaba |
| Service_State | char(25) | Service state | Lagos |

## BOOKING

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Booking_Id(PK) | int | Booking's identifier | 3219 |
| Service_Id(FK) | int | Service identifier | 321 |
| User_Id(FK) | Int | Service client | 432 |
| Booking_Date | datetime | Booking's date | 31/03/2021 09:30:00AM |

## TASK

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Task_Id(PK) | int | Task's identifier | 8983 |
| Task_Description | varchar(1000) | Task's description | I need a french language tutor |
| Task_Price | double | Amount to be paid | 4257.99 |
| Task_Status | char(10) | Task's status | Opened/Closed |
| Task_Expiration_Date | datetime | Application deadline | 23/03/2015 03:30:00AM |

## OFFER

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Offer_Id(PK) | int | Offer's identifier | 8983 |
| Task_Id(FK) | int | Task's identifier | 6782 |
| User_Id(FK) | int | User making an offer | 3 |
| Offer_Description | varchar(1000) | Offer's description | Je ne sais quois, va va voom!!, |

| | | | |
|---|---|---|---|
| Offer_Price | double | Asking price | 50000.00 |
| Offer_Status | char(10) | Task’s status | Accepted/Pending/ Rejected |
| Offer_Timestamp | datetime | Offer’s timestamp | 23/03/2015 03:30:00AM |

# REVIEW

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Review_Id(PK) | int | Review’s identifier | 8983 |
| User_Id(FK) | int | Reviewer’s identifier | 6782 |
| Shop_Id(FK) | int(optional) | Shop being reviewed | 3 |
| Service_Id(FK) | int(optional) | Service being reviewed | 43 |
| Review_Comment | varchar(1000) | Reviewer’s comment | Very very poor. Caused an electrical fire in my home |
| Review_Rating | double | Reviewer’s rating | 3.5 |
| Review_Timestamp | datetime | Review's timestamp | 23/03/2015 03:30:00AM |

# PAYMENT

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Payment_Id(PK) | int | Payment’s identifier | 8983 |
| User_Id(FK) | int | User making payment’s identifier | 6782 |
| Order_Id(FK) | int(optional) | Order’s identifier | 3 |
| Booking_Id(FK) | int(optional) | Booking’s identifier | 43 |
| Payment_Amount | double | Amount paid | -3293.00 |
| Payment_Reference | char(25) | Payment reference | TGNX27392209ZG |
| Payment_Type | char(25) | Payment type | Deposit/Withdrawal/ Shop/Service |
| Payment_Timestamp | datetime | Payment’s timestamp | 23/03/2015 03:30:00AM |

# CATEGORY

| Attributes | Data Type | Description | Example |
|---|---|---|---|
| Category_Type | char(25) | Category type | Electronics |
| Task_Id(FK) | int(optional) | Task's being categorised | 6782 |
| Item_Id(FK) | int(optional) | Item being categorised | 3 |
| Service_Id(FK) | int(optional) | Service being categorised | 43 |

# 4. Implementation

## 4.1. SQL Code

```
CREATE TABLE `Taaskly`.`USER` (
  `User_Id` INT NOT NULL,
  `User_Email` CHAR(50) NOT NULL,
  `User_Nickname` CHAR(25) NOT NULL,
  `User_Firstname` CHAR(25) NOT NULL,
  `User_Lastname` CHAR(25) NOT NULL,
  `User_Bio` VARCHAR(1000) NULL,
  `User_Phone` CHAR(13) NOT NULL,
  `User_Balance` DOUBLE NOT NULL,
  `User_Referral_Code` CHAR(6) NOT NULL,
  `User_Profile_Picture` BLOB NULL,
  PRIMARY KEY (`User_Id`));

CREATE TABLE `Taaskly`.`SHOP` (
  `Shop_Id` INT NOT NULL,
  `Shop_Name` CHAR(50) NOT NULL,
  `Shop_Profile_Picture` BLOB NOT NULL,
  `Shop_Bio` VARCHAR(1000) NOT NULL,
  `Shop_Phone` CHAR(13) NOT NULL,
  `Shop_Street` CHAR(25) NOT NULL,
  `Shop_City` CHAR(25) NOT NULL,
  `Shop_State` CHAR(25) NOT NULL,
  PRIMARY KEY (`Shop_Id`));

CREATE TABLE `Taaskly`.`ITEM` (
  `Item_Id` INT NOT NULL,
  `Shop_Id` INT NOT NULL,
  `Item_Name` CHAR(50) NOT NULL,
  `Item_Description` VARCHAR(1000) NOT NULL,
  `Item_Price` DOUBLE NOT NULL,
  `Item_Image` BLOB NOT NULL,
  PRIMARY KEY (`Item_Id`),
  INDEX `fk_ITEM_1_idx` (`Shop_Id` ASC) VISIBLE,
  CONSTRAINT `fk_ITEM_1`
    FOREIGN KEY (`Shop_Id`)
    REFERENCES `Taaskly`.`SHOP` (`Shop_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `Taaskly`.`SERVICE` (
  `Service_Id` INT NOT NULL,
  `Service_Name` CHAR(25) NOT NULL,
```

```
  `Service_Bio` VARCHAR(1000) NOT NULL,
  `Service_Phone` CHAR(13) NOT NULL,
  `Service_Profile_Picture` BLOB NOT NULL,
  `Service_Street` CHAR(25) NOT NULL,
  `Service_City` CHAR(25) NOT NULL,
  `Service_State` CHAR(25) NOT NULL,
  PRIMARY KEY (`Service_Id`));

CREATE TABLE `Taaskly`.`TASK` (
  `Task_Id` INT NOT NULL,
  `Task_Description` VARCHAR(1000) NOT NULL,
  `Task_Price` DOUBLE NOT NULL,
  `Task_Status` VARCHAR(45) NOT NULL,
  `Task_Expiration_Date` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Task_Id`));

CREATE TABLE `Taaskly`.`REVIEW` (
  `Review_Id` INT NOT NULL,
  `User_Id` INT NOT NULL,
  `Shop_Id` INT NULL,
  `Service_Id` INT NULL,
  `Review_Comment` VARCHAR(1000) NOT NULL,
  `Review_Rating` DOUBLE NOT NULL,
  `Review_Timestamp` DATETIME NOT NULL,
  PRIMARY KEY (`Review_Id`),
  INDEX `fk_REVIEW_1_idx` (`User_Id` ASC) VISIBLE,
  INDEX `fk_REVIEW_2_idx` (`Shop_Id` ASC) VISIBLE,
  INDEX `fk_REVIEW_3_idx` (`Service_Id` ASC) VISIBLE,
  CONSTRAINT `fk_REVIEW_1`
    FOREIGN KEY (`User_Id`)
    REFERENCES `Taaskly`.`USER` (`User_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_REVIEW_2`
    FOREIGN KEY (`Shop_Id`)
    REFERENCES `Taaskly`.`SHOP` (`Shop_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_REVIEW_3`
    FOREIGN KEY (`Service_Id`)
    REFERENCES `Taaskly`.`SERVICE` (`Service_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

```
CREATE TABLE `Taaskly`.`CATEGORY` (
  `Category_Type` CHAR(25) NOT NULL,
  `Task_Id` INT NULL,
  `Item_Id` INT NULL,
  `Service_Id` INT NULL,
  INDEX `fk_CATEGORY_1_idx` (`Task_Id` ASC) VISIBLE,
  INDEX `fk_CATEGORY_2_idx` (`Item_Id` ASC) VISIBLE,
  INDEX `fk_CATEGORY_3_idx` (`Service_Id` ASC) VISIBLE,
  CONSTRAINT `fk_CATEGORY_1`
    FOREIGN KEY (`Task_Id`)
    REFERENCES `Taaskly`.`TASK` (`Task_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_CATEGORY_2`
    FOREIGN KEY (`Item_Id`)
    REFERENCES `Taaskly`.`ITEM` (`Item_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_CATEGORY_3`
    FOREIGN KEY (`Service_Id`)
    REFERENCES `Taaskly`.`SERVICE` (`Service_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `Taaskly`.`OFFER` (
  `Offer_Id` INT NOT NULL,
  `Task_Id` INT NOT NULL,
  `User_Id` INT NOT NULL,
  `Offer_Description` VARCHAR(1000) NOT NULL,
  `Offer_Price` DOUBLE NOT NULL,
  `Offer_Status` CHAR(10) NOT NULL,
  `Offer_Timestamp` DATETIME NOT NULL,
  PRIMARY KEY (`Offer_Id`),
  INDEX `fk_OFFER_1_idx` (`User_Id` ASC) VISIBLE,
  INDEX `fk_OFFER_2_idx` (`Task_Id` ASC) VISIBLE,
  CONSTRAINT `fk_OFFER_1`
    FOREIGN KEY (`User_Id`)
    REFERENCES `Taaskly`.`USER` (`User_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_OFFER_2`
    FOREIGN KEY (`Task_Id`)
    REFERENCES `Taaskly`.`TASK` (`Task_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

```
CREATE TABLE `Taaskly`.`BOOKING` (
  `Booking_Id` INT NOT NULL,
  `Service_Id` INT NOT NULL,
  `User_Id` INT NOT NULL,
  `Booking_Date` DATETIME NOT NULL,
  PRIMARY KEY (`Booking_Id`),
  INDEX `fk_BOOKING_1_idx` (`Service_Id` ASC) VISIBLE,
  INDEX `fk_BOOKING_2_idx` (`User_Id` ASC) VISIBLE,
  CONSTRAINT `fk_BOOKING_1`
    FOREIGN KEY (`Service_Id`)
    REFERENCES `Taaskly`.`SERVICE` (`Service_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_BOOKING_2`
    FOREIGN KEY (`User_Id`)
    REFERENCES `Taaskly`.`USER` (`User_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `Taaskly`.`ORDER` (
  `Order_Id` INT NOT NULL,
  `Shop_Id` INT NOT NULL,
  `User_Id` INT NOT NULL,
  `Order_Cost` DOUBLE NOT NULL,
  PRIMARY KEY (`Order_Id`),
  INDEX `fk_ORDER_1_idx` (`Shop_Id` ASC) VISIBLE,
  INDEX `fk_ORDER_2_idx` (`User_Id` ASC) VISIBLE,
  CONSTRAINT `fk_ORDER_1`
    FOREIGN KEY (`Shop_Id`)
    REFERENCES `Taaskly`.`SHOP` (`Shop_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ORDER_2`
    FOREIGN KEY (`User_Id`)
    REFERENCES `Taaskly`.`USER` (`User_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `Taaskly`.`ORDER_LINE` (
  `Order_Line_Id` INT NOT NULL,
  `Order_Id` INT NOT NULL,
  `Item_Id` INT NOT NULL,
  `Order_Line_Item_Quantity` INT NOT NULL,
  `Order_Line_Cost` DOUBLE NOT NULL,
  PRIMARY KEY (`Order_Line_Id`),
```

```
  INDEX `fk_ORDER_LINE_1_idx` (`Order_Id` ASC) VISIBLE,
  INDEX `fk_ORDER_LINE_2_idx` (`Item_Id` ASC) VISIBLE,
  CONSTRAINT `fk_ORDER_LINE_1`
    FOREIGN KEY (`Order_Id`)
    REFERENCES `Taaskly`.`ORDER` (`Order_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ORDER_LINE_2`
    FOREIGN KEY (`Item_Id`)
    REFERENCES `Taaskly`.`ITEM` (`Item_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

CREATE TABLE `Taaskly`.`PAYMENT` (
  `Payment_Id` INT NOT NULL,
  `User_Id` INT NOT NULL,
  `Order_Id` INT NULL,
  `Booking_Id` INT NULL,
  `Payment_Amount` DOUBLE NOT NULL,
  `Payment_Reference` CHAR(25) NOT NULL,
  `Payment_Type` CHAR(25) NOT NULL,
  `Payment_Timestamp` DATETIME NOT NULL,
  PRIMARY KEY (`Payment_Id`),
  INDEX `fk_PAYMENT_1_idx` (`User_Id` ASC) VISIBLE,
  INDEX `fk_PAYMENT_2_idx` (`Order_Id` ASC) VISIBLE,
  INDEX `fk_PAYMENT_3_idx` (`Booking_Id` ASC) VISIBLE,
  CONSTRAINT `fk_PAYMENT_1`
    FOREIGN KEY (`User_Id`)
    REFERENCES `Taaskly`.`USER` (`User_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_PAYMENT_2`
    FOREIGN KEY (`Order_Id`)
    REFERENCES `Taaskly`.`ORDER` (`Order_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_PAYMENT_3`
    FOREIGN KEY (`Booking_Id`)
    REFERENCES `Taaskly`.`BOOKING` (`Booking_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```