Bike Rental Shop Project

```bash
#!/bin/bash

PSQL="psql -X --username=freecodecamp --dbname=bikes --tuples-only -c"

echo -e "\n~~~~~ Bike Rental Shop ~~~~~\n"

MAIN_MENU() {
  if [[ $1 ]]
  then
    echo -e "\n$1"
  fi

  echo "How may I help you?"
  echo -e "\n1. Rent a bike\n2. Return a bike\n3. Exit"
  read MAIN_MENU_SELECTION

  case $MAIN_MENU_SELECTION in
    1) RENT_MENU ;;
    2) RETURN_MENU ;;
    3) EXIT ;;
    *) MAIN_MENU "Please enter a valid option." ;;
  esac
}

RENT_MENU() {
  # get available bikes
  AVAILABLE_BIKES=$($PSQL "SELECT bike_id, type, size FROM bikes WHERE available = true ORDER BY bike_id")

  # if no bikes available
  if [[ -z $AVAILABLE_BIKES ]]
  then
    # send to main menu
    MAIN_MENU "Sorry, we don't have any bikes available right now."
  else
    # display available bikes
    echo -e "\nHere are the bikes we have available:"
    echo "$AVAILABLE_BIKES" | while read BIKE_ID BAR TYPE BAR SIZE
    do
```

```
      echo "$BIKE_ID) $SIZE\" $TYPE Bike"
    done

    # ask for bike to rent
    echo -e "\nWhich one would you like to rent?"
    read BIKE_ID_TO_RENT

    # if input is not a number
    if [[ ! $BIKE_ID_TO_RENT =~ ^[0-9]+$ ]]
    then
      # send to main menu
      MAIN_MENU "That is not a valid bike number."
    else
      # get bike availability
      BIKE_AVAILABILITY=$($PSQL "SELECT available FROM bikes WHERE bike_id =
$BIKE_ID_TO_RENT AND available = true")


      # if not available
      if [[ -z $BIKE_AVAILABILITY ]]
      then
        # send to main menu
        MAIN_MENU "That bike is not available."
      else
        # get customer info
        echo -e "\nWhat's your phone number?"
        read PHONE_NUMBER
        CUSTOMER_NAME=$($PSQL "SELECT name FROM customers WHERE phone =
'$PHONE_NUMBER'")
        # if customer doesn't exist
        if [[ -z $CUSTOMER_NAME ]]
        then
          # get new customer name
          echo -e "\nWhat's your name?"
          read CUSTOMER_NAME
          # insert new customer
          INSERT_CUSTOMER_RESULT=$($PSQL "INSERT INTO customers(name, phone)
VALUES('$CUSTOMER_NAME', '$PHONE_NUMBER')")
        fi
        # get customer_id
        CUSTOMER_ID=$($PSQL "SELECT customer_id FROM customers WHERE phone =
'$PHONE_NUMBER'")
        # insert bike rental
        INSERT_RENTAL_RESULT=$($PSQL "INSERT INTO rentals(customer_id, bike_id) VALUES($CUSTOMER_ID, $BIKE_ID_TO_RENT)")
        # set bike availability to false
        SET_TO_FALSE_RESULT=$($PSQL "UPDATE bikes SET available = false WHERE bike_id = $BIKE_ID_TO_RENT")
        # get bike info
        BIKE_INFO=$($PSQL "SELECT size, type FROM bikes WHERE bike_id = $BIKE_ID_TO_RENT")
        BIKE_INFO_FORMATTED=$(echo $BIKE_INFO | sed 's/ |/"/')
        # send to main menu
        MAIN_MENU "I have put you down for the $BIKE_INFO_FORMATTED Bike, $(echo $CUSTOMER_NAME | sed -r 's/^ *| *$//g')."
      fi
    fi
  fi
}

RETURN_MENU() {
  # get customer info
  echo -e "\nWhat's your phone number?"
  read PHONE_NUMBER
  CUSTOMER_ID=$($PSQL "SELECT customer_id FROM customers WHERE phone = '$PHONE_NUMBER'")
  # if not found
  if [[ -z $CUSTOMER_ID ]]
  then
    # send to main menu
    MAIN_MENU "I could not find a record for that phone number."
  else
    # get customer's rentals
    CUSTOMER_RENTALS=$($PSQL "SELECT bike_id, type, size FROM bikes INNER JOIN rentals USING(bike_id) INNER JOIN customers USING(customer_id) WHERE phone = '$PHONE_NUMBER' AND date_returned IS NULL ORDER BY bike_id")

    # if no rentals
    if [[ -z $CUSTOMER_RENTALS ]]
    then
      # send to main menu
      MAIN_MENU "You do not have any bikes rented."
    else
      # display rented bikes
      echo -e "\nHere are your rentals:"
      echo "$CUSTOMER_RENTALS" | while read BIKE_ID BAR TYPE BAR SIZE
      do
        echo "$BIKE_ID) $SIZE\" $TYPE Bike"
      done
      # ask for bike to return
      echo -e "\nWhich one would you like to return?"
      read BIKE_ID_TO_RETURN
      # if not a number
      if [[ ! $BIKE_ID_TO_RETURN =~ ^[0-9]+$ ]]
      then
        # send to main menu
        MAIN_MENU "That is not a valid bike number."
      else
        # check if input is rented
        RENTAL_ID=$($PSQL "SELECT rental_id FROM rentals INNER JOIN customers
USING(customer_id) WHERE phone = '$PHONE_NUMBER' AND bike_id =
$BIKE_ID_TO_RETURN AND date_returned IS NULL")
        # if input not rented
        if [[ -z $RENTAL_ID ]]
        then
          # send to main menu
          MAIN_MENU "You do not have that bike rented."
        else
          # update date_returned
          RETURN_BIKE_RESULT=$($PSQL "UPDATE rentals SET date_returned = NOW()
WHERE rental_id = $RENTAL_ID ")
          # set bike availability to true
          SET_TO_TRUE_RESULT=$($PSQL "UPDATE bikes SET available = true WHERE
bike_id = $BIKE_ID_TO_RETURN")
          # send to main menu
          MAIN_MENU "Thank you for returning your bike."
        fi

      fi

    fi

  fi
}
EXIT() {
  echo -e "\nThank you for stopping in.\n

MAIN_MENU
```