AC51049 - Database Systems Development
Coursework 1 - Cover Sheet

TEAM NUMBER **:**11
COMPANY NAME **:** BUNIQ Co. Ltd
COMPANY SLOGAN **:** Be Bold, Be Unique, BUNIQ

TEAM MEMBERS : Aditya Nikumbh
: Rakshith Ramureddy
: Quoc Tran
: Benjawan Hommak
: Divya Parmar
: Hung Tran
: Panuchart Laorungrat
: Rohit Advani
: Roseline Chibuzor Inyamah
: Abiola Ayinla

✓ **Company description / Specification report**
Word count (900-1000): 901 **Number of pages** **:** 4
✓ **E-R diagram** **Number of pages** **:** 1
✓ **SQL CREATE statements** **Number of pages** **:** 9

Total pages **:** 14

✓ **We confirm that the team members have read and understood the University policy on Academic Misconduct**

BUNIQ Co. Ltd.

Company profile:

At BUNIQ Co. Ltd., we take immense pride in being a stalwart in the United Kingdom's vibrant sports retailing scene. With a remarkable journey spanning well over a decade, we have established ourselves as a pillar of excellence and dedication in athletics.

Our journey in the UK sports industry began with a vision to offer a comprehensive range of solutions and services that meet the diverse needs of athletes at every level. We've forged strong partnerships with local and international suppliers, carefully curating our product range to exceed customer expectations. BUNIQ Co. Ltd. aims to inspire, equip, and connect with the sporting community, dedicated to enhancing lives through active, healthy living, ensuring sports accessibility for all, and upholding ethical and sustainable practices. Our legacy is one of excellence, primarily focusing on our valued customers.

In Harmony with Our Mission

At BUNIQ Co. Ltd., our mission is straightforward: we exist to provide you with top-tier sports gear and services. We aspire to inspire, equip, and foster connections within the sporting community on a profound level. Our legacy within the UK's sports arena is a testament to our unwavering dedication to excellence, anchored by our steadfast commitment to placing you, our esteemed customers, at the centre of everything we do.

Understanding Our Users

To steer our design project in the right direction, it's paramount that we comprehend the roles of four pivotal user types who interact with BUNIQ across various touchpoints:

- **Online Shoppers**:
  - These tech-savvy customers explore our website to hunt for

products, make digital purchases, and seek timely assistance through online channels.
    - The design must ensure an intuitive online shopping experience with secure payment gateways and seamless navigation.
- **In-Store Shoppers**:
    - Our in-store customers are the ones who physically grace our branches, expecting an immersive and delightful shopping journey.
    - Design considerations should encompass physical store layouts, clear product signage, and an atmosphere that fosters a positive in-store customer experience.
- **Suppliers**:
    - Suppliers are the unsung heroes in our supply chain, diligently ensuring that our shelves are always stocked with high-quality products.
    - The design project should facilitate streamlined communication and order management, simplifying the supplier experience for both online and in-store operations.
- **Employees**:
    - Our dedicated team members handle various facets of our business, from inventory control to customer service and maintaining our robust database.
    - The design should incorporate systems that make inventory management efficient, streamline customer service, and maintain meticulous financial records.

Our Database and Policies

Our secure and meticulously organised database is the engine that powers BUNIQ's operations, which is indispensable for online and in-store customer journeys. This database serves as the backbone for various policies and procedures that ensure the company's efficiency and integrity:

Constraints:

- **Data Security and Privacy**: Your privacy is paramount at BUNIQ Co. Ltd. We employ stringent measures, including encryption, and restrict access to authorised team members to safeguard your data.
- **Data Accuracy**: We take pride in maintaining up-to-date, error-free information in our database, ensuring you have complete confidence in the data you find with us.

- **Supplier Relations**: Our commitment to strong and ethical supplier relationships is reflected in our policies. We enforce fair practices, on-time payments, and responsible sourcing.
- **Inventory Management**: With vigilant inventory oversight, we ensure the correct product levels, thanks to our database, preventing overstocking and product shortages.
- **Customer Service**: Your satisfaction matters. Our database supports excellent customer service, guiding us in assisting you efficiently and effectively.
- **Financial Accountability**: Your trust is our responsibility. Our database meticulously tracks every payment, invoice, and receipt for transparency and accuracy.

Policy Highlights**:**

- **Data Retention**: We adhere to a data retention policy, only keeping data as necessary, based on regulations and business needs, and regularly purging outdated or unnecessary data.
- **Supplier Code of Conduct**: Our suppliers, considered part of the BUNIQ family, adhere to a Supplier Code of Conduct, reinforcing ethical expectations through our database.
- **Inventory Control**: Our database ensures inventory control, maintaining product availability while avoiding overstocking or shortages of your preferred items.
- **Customer Privacy**: We protect your data with a Customer Privacy Policy, ensuring only authorised personnel can access your information and seek your consent before using your data.
- **Financial Transparency**: Our database is a pivotal tool in upholding financial accountability and transparency, meticulously recording payments, invoices, and receipts for precise and accurate financial records.
- **Data Access**: Your data's security is our priority. We strictly control access, permitting it only to authorised individuals with a genuine need. Access is monitored and tracked to maintain data safety and privacy.

Conclusion:

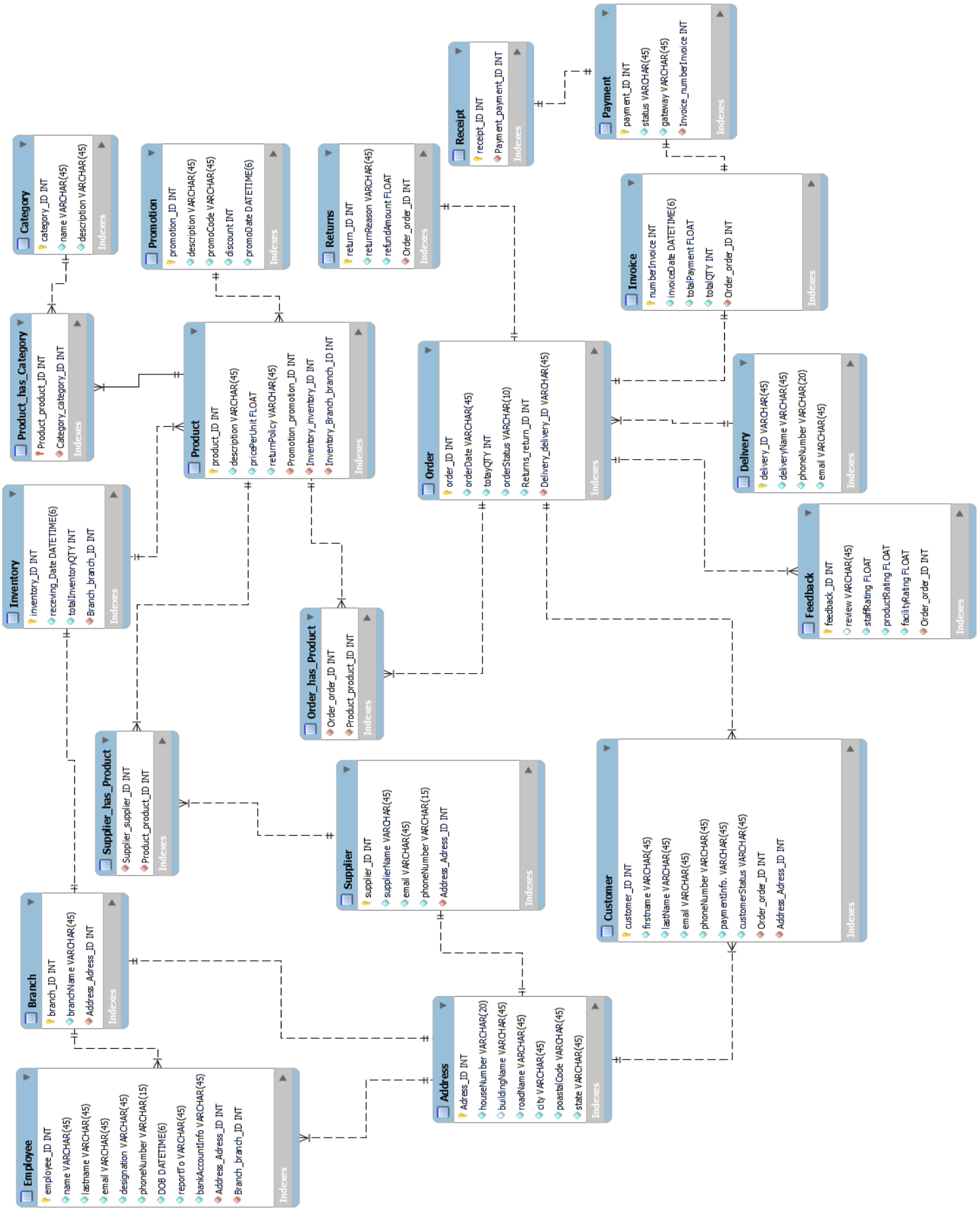
BUNIQ, the UK's most prominent sports retailer, is unwavering in its devotion to fundamental values such as quality, responsibility, and customer happiness. Our commitment extends beyond providing outstanding sports equipment, including ethical business practises and a deep commitment to the broader athletic

community.

We take great pleasure in our services, designed to ensure that each customer's experience is fantastic. We're more than a sports company; we're your dedicated partner in the quest for athletic excellence and the benefit of the sporting community, from the moment you choose our products to the assistance you receive after your purchase.

At BUNIQ, we are more than just a company; we are your partner in pursuing athletic excellence, environmental responsibility, and community betterment. Our quest is more than just a business venture; it is a holistic commitment to enhancing the world of sports and generating good change.

Entity-Relationship Diagram



SQL Statements

```
-- MySQL Script generated by MySQL Workbench
-- Wed Oct 18 15:24:59 2023
-- Model: New Model    Version: 1.0
-- 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_DAT
E,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTIO
N';

-- -----------------------------------------------------
-- Schema mydb1
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema mydb1
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb1` DEFAULT CHARACTER SET utf8 ;
USE `mydb1` ;

-- -----------------------------------------------------
-- Table `mydb1`.`Address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Address` (
  `Adress_ID` INT NOT NULL,
  `houseNumber` VARCHAR(20) NOT NULL,
  `buildingName` VARCHAR(45) NULL,
  `roadName` VARCHAR(45) NOT NULL,
  `city` VARCHAR(45) NOT NULL,
```

```
  `poastalCode` VARCHAR(45) NOT NULL,
  `state` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Adress_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Branch`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Branch` (
  `branch_ID` INT NOT NULL,
  `branchName` VARCHAR(45) NOT NULL,
  `Address_Adress_ID` INT NOT NULL,
  PRIMARY KEY (`branch_ID`),
    FOREIGN KEY (`Address_Adress_ID`)
    REFERENCES `mydb1`.`Address` (`Adress_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Employee` (
  `employee_ID` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `lastname` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `designation` VARCHAR(45) NOT NULL,
  `phoneNumber` VARCHAR(15) NOT NULL,
  `DOB` DATETIME(6) NOT NULL,
  `reportTo` VARCHAR(45) NOT NULL,
  `bankAccountInfo` VARCHAR(45) NOT NULL,
  `Address_Adress_ID` INT NOT NULL,
  `Branch_branch_ID` INT NOT NULL,
  PRIMARY KEY (`employee_ID`),
    FOREIGN KEY (`Address_Adress_ID`)
```

```
    REFERENCES `mydb1`.`Address` (`Adress_ID`),
    FOREIGN KEY (`Branch_branch_ID`)
    REFERENCES `mydb1`.`Branch` (`branch_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Supplier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Supplier` (
  `supplier_ID` INT NOT NULL,
  `supplierName` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phoneNumber` VARCHAR(15) NOT NULL,
  `Address_Adress_ID` INT NOT NULL,
  PRIMARY KEY (`supplier_ID`),
  INDEX `fk_Supplier_Address1_idx` (`Address_Adress_ID` ASC) VISIBLE,
    FOREIGN KEY (`Address_Adress_ID`)
    REFERENCES `mydb1`.`Address` (`Adress_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Delivery`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Delivery` (
  `delivery_ID` VARCHAR(45) NOT NULL,
  `deliveryName` VARCHAR(45) NOT NULL,
  `phoneNumber` VARCHAR(20) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`delivery_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Order`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `mydb1`.`Order` (
  `order_ID` INT NOT NULL,
  `orderDate` VARCHAR(45) NOT NULL,
  `totayQTY` INT NOT NULL,
  `orderStatus` VARCHAR(10) NOT NULL,
  `Returns_return_ID` INT NOT NULL,
  `Delivery_delivery_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`order_ID`),
    FOREIGN KEY (`Delivery_delivery_ID`)
    REFERENCES `mydb1`.`Delivery` (`delivery_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Customer` (
  `customer_ID` INT NOT NULL,
  `firstname` VARCHAR(45) NOT NULL,
  `lastName` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phoneNumber` VARCHAR(45) NOT NULL,
  `paymentInfo.` VARCHAR(45) NOT NULL,
  `customerStatus` VARCHAR(45) NOT NULL,
  `Order_order_ID` INT NOT NULL,
  `Address_Adress_ID` INT NOT NULL,
  PRIMARY KEY (`customer_ID`),
    FOREIGN KEY (`Order_order_ID`)
    REFERENCES `mydb1`.`Order` (`order_ID`),
    FOREIGN KEY (`Address_Adress_ID`)
    REFERENCES `mydb1`.`Address` (`Adress_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Promotion`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Promotion` (
  `promotion_ID` INT NOT NULL,
  `description` VARCHAR(45) NOT NULL,
  `promoCode` VARCHAR(45) NOT NULL,
  `discount` INT NOT NULL,
  `promoDate` DATETIME(6) NOT NULL,
  PRIMARY KEY (`promotion_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Inventory`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Inventory` (
  `inventory_ID` INT NOT NULL,
  `receving_Date` DATETIME(6) NOT NULL,
  `totalInventoryQTY` INT NOT NULL,
  `Branch_branch_ID` INT NOT NULL,
  PRIMARY KEY (`inventory_ID`),
    FOREIGN KEY (`Branch_branch_ID`)
    REFERENCES `mydb1`.`Branch` (`branch_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Product` (
  `product_ID` INT NOT NULL,
  `description` VARCHAR(45) NOT NULL,
  `pricePerUnit` FLOAT NOT NULL,
  `returnPolicy` VARCHAR(45) NOT NULL,
  `Promotion_promotion_ID` INT NOT NULL,
  `Inventory_inventory_ID` INT NOT NULL,
  `Inventory_Branch_branch_ID` INT NOT NULL,
```

```
  PRIMARY KEY (`product_ID`),
    FOREIGN KEY (`Promotion_promotion_ID`)
    REFERENCES `mydb1`.`Promotion` (`promotion_ID`));
   -- FOREIGN KEY (`Inventory_inventory_ID` , `Inventory_Branch_branch_ID`)
   --  REFERENCES `mydb1`.`Inventory` (`inventory_ID` , `Branch_branch_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Invoice`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Invoice` (
  `numberInvoice` INT NOT NULL,
  `invoiceDate` DATETIME(6) NOT NULL,
  `totalPayment` FLOAT NOT NULL,
  `totalQTY` INT NOT NULL,
  `Order_order_ID` INT NOT NULL,
  PRIMARY KEY (`numberInvoice`),
    FOREIGN KEY (`Order_order_ID`)
    REFERENCES `mydb1`.`Order` (`order_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Feedback`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Feedback` (
  `feedback_ID` INT NOT NULL,
  `review` VARCHAR(45) NULL,
  `staffRating` FLOAT NOT NULL,
  `productRating` FLOAT NOT NULL,
  `facilityRating` FLOAT NOT NULL,
  `Order_order_ID` INT NOT NULL,
  PRIMARY KEY (`feedback_ID`),
    FOREIGN KEY (`Order_order_ID`)
    REFERENCES `mydb1`.`Order` (`order_ID`));
```

```
-- -----------------------------------------------------
-- Table `mydb1`.`Payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Payment` (
  `payment_ID` INT NOT NULL,
  `status` VARCHAR(45) NOT NULL,
  `gateway` VARCHAR(45) NOT NULL,
  `Invoice_numberInvoice` INT NOT NULL,
  PRIMARY KEY (`payment_ID`),
    FOREIGN KEY (`Invoice_numberInvoice`)
    REFERENCES `mydb1`.`Invoice` (`numberInvoice`));


-- -----------------------------------------------------
-- Table `mydb1`.`Returns`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Returns` (
  `return_ID` INT NOT NULL,
  `returnReason` VARCHAR(45) NOT NULL,
  `refundAmount` FLOAT NOT NULL,
  `Order_order_ID` INT NOT NULL,
  PRIMARY KEY (`return_ID`),
    FOREIGN KEY (`Order_order_ID`)
    REFERENCES `mydb1`.`Order` (`order_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Category`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Category` (
  `category_ID` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
```

```
  `description` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`category_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Product_has_Category`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Product_has_Category` (
  `Product_product_ID` INT NOT NULL,
  `Category_category_ID` INT NOT NULL,
  PRIMARY KEY (`Product_product_ID`),
    FOREIGN KEY (`Product_product_ID`)
    REFERENCES `mydb1`.`Product` (`product_ID`),
    FOREIGN KEY (`Category_category_ID`)
    REFERENCES `mydb1`.`Category` (`category_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Receipt`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Receipt` (
  `receipt_ID` INT NOT NULL,
  `Payment_payment_ID` INT NOT NULL,
  PRIMARY KEY (`receipt_ID`),
    FOREIGN KEY (`Payment_payment_ID`)
    REFERENCES `mydb1`.`Payment` (`payment_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Supplier_has_Product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Supplier_has_Product` (
  `Supplier_supplier_ID` INT NOT NULL,
  `Product_product_ID` INT NOT NULL,
```

```
    FOREIGN KEY (`Supplier_supplier_ID`)
    REFERENCES `mydb1`.`Supplier` (`supplier_ID`),
    FOREIGN KEY (`Product_product_ID`)
    REFERENCES `mydb1`.`Product` (`product_ID`));


-- -----------------------------------------------------
-- Table `mydb1`.`Order_has_Product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb1`.`Order_has_Product` (
  `Order_order_ID` INT NOT NULL,
  `Product_product_ID` INT NOT NULL,
    FOREIGN KEY (`Order_order_ID`)
    REFERENCES `mydb1`.`Order` (`order_ID`),
    FOREIGN KEY (`Product_product_ID`)
    REFERENCES `mydb1`.`Product` (`product_ID`));


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