Database Creation

1. Requirements Analysis

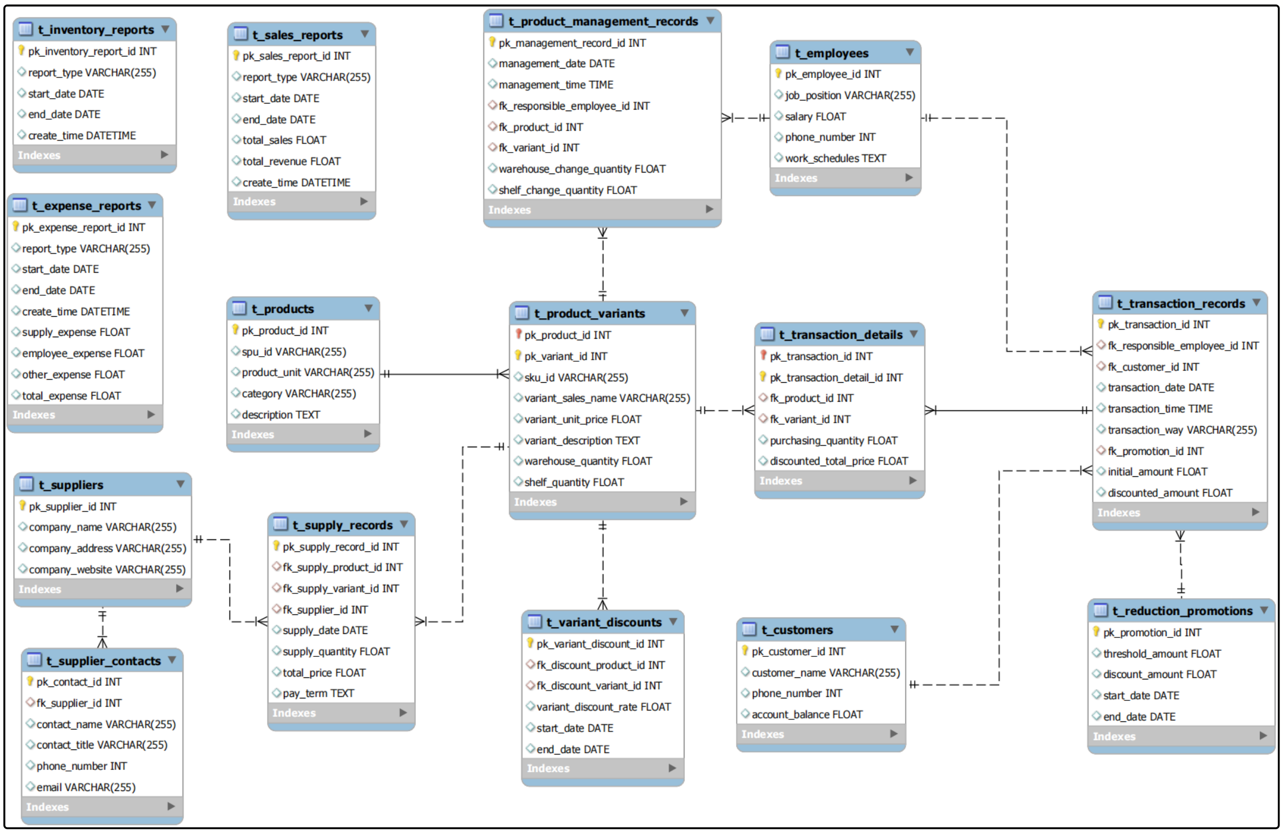
1.1. Comprehensive Supermarket Requirements

The database for the comprehensive supermarket need basically five subsystems, which are:

1. **Product Organization System:** Since a supermarket will sell a lot of products, it needs to be able to accurately identify each item.
2. **Product Supply System:** A supermarket must have a source of supply for its products.
3. **Product Sales System:** A supermarket needs to sell the products, hence need to have a large system for transaction.
4. **Employee Management System:** A supermarket needs employees hence need a system to manage them.
5. **Marketing Record System:** A supermarket needs to know how much it earns for selling, and how much it spends for maintaining this supermarket.

2. Implementation of Database

2.1. Overview Structure



2.2. Details about Product Organization System

2.2.1. t_products (以SPU为单位的entity)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_products` (
  `pk_product_id` INT NOT NULL,
  `spu_id` VARCHAR(255) NULL DEFAULT NULL,
  `product_unit` VARCHAR(255) NULL DEFAULT NULL,
  `category` VARCHAR(255) NULL DEFAULT NULL,
  `description` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `pk_product_id` - PK, 标识每一样产品的SPU层面编号, 例如 "001", "002".
- `spu_id` - SPU (Standard Product Unit), 标准产品单位, 例如 "HUAWEI P50 Pro".

- `product_unit` - 商品的计量单位，例如 "kilogram", "item".
- `category` - 商品的分类区，例如 "Groceries", "Personal Care & Beauty".
- `description` - 对商品的描述，例如 "HUAWEI P50 Pro is a high-end smartphone launched by Huawei. It is part of the Huawei P series, which is known for its high-quality photography capabilities."

2.2.2. t_product_variants (以SKU为单位的entity)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_product_variants` (
  `pk_product_id` INT NOT NULL,
  `pk_variant_id` INT NOT NULL,
  `sku_id` VARCHAR(255) NULL DEFAULT NULL,
  `variant_sales_name` VARCHAR(255) NULL DEFAULT NULL,
  `variant_unit_price` FLOAT NULL DEFAULT NULL,
  `variant_description` TEXT NULL DEFAULT NULL,
  `warehouse_quantity` FLOAT NULL DEFAULT NULL,
  `shelf_quantity` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_product_id`, `pk_variant_id`),
  CONSTRAINT `t_product_variants_ibfk_1`
    FOREIGN KEY (`pk_product_id`)
    REFERENCES `comprehensive_supermarket`.`t_products` (`pk_product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `pk_product_id` - PK, 标识每一样产品的SPU层面编号, 例如 "001", "002". 同时也是FK, 指向表 `t_products` 中的 `pk_product_id`. 因此该实体是一个依赖于 `t_products` 的弱实体。
- `pk_variant_id` - PK, 标识每一件产品变体的SKU层面编号，例如 "001", "002".
- `sku_id` - SKU (Stock Keeping Unit), 库存管理单位, 例如 "HUAWEI_P50_P01" 是华为P50Pro炫彩天空蓝的编号, 而 "HUAWEI_P50_P02" 是华为P50Pro冷静流潋紫的编号. 该单位以每一种商品的变体为单位，用库存表示数量.
- `variant_sales_name` - 商品变种的销售名称，例如 "HUAWEI P50 Pro Sky-blue Fantastic". 该销售名称为货架上的销售名称，而 `sku_id` 则更像是一个编号.
- `variant_unit_price` - 单价.
- `variant_description` - 对该产品变体的具体描述，一般为特殊备注.
- `warehouse_quantity` - 仓库库存量.
- `shelf_quantity` - 货架库存量.

> 之所以库存量的数值类型为 `float`, 是因为可能要处理类似于猪肉这样按斤为单位卖的货物.

2.2.3. t_product_management_records (产品管理记录)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_product_management_records` (
  `pk_management_record_id` INT NOT NULL,
  `management_date` DATE NULL DEFAULT NULL,
  `management_time` TIME NULL DEFAULT NULL,
  `fk_responsible_employee_id` INT NULL DEFAULT NULL,
  `fk_product_id` INT NULL DEFAULT NULL,
  `fk_variant_id` INT NULL DEFAULT NULL,
  `warehouse_change_quantity` FLOAT NULL DEFAULT NULL,
  `shelf_change_quantity` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_management_record_id`),
  INDEX `fk_responsible_employee_id` (`fk_responsible_employee_id` ASC) VISIBLE,
  INDEX `fk_product_id` (`fk_product_id` ASC, `fk_variant_id` ASC) VISIBLE,
  CONSTRAINT `product_management_records_ibfk_1`
    FOREIGN KEY (`fk_responsible_employee_id`)
    REFERENCES `comprehensive_supermarket`.`t_employees` (`pk_employee_id`),
  CONSTRAINT `product_management_records_ibfk_2`
    FOREIGN KEY (`fk_product_id` , `fk_variant_id`)
    REFERENCES `comprehensive_supermarket`.`t_product_variants` (`pk_product_id` , `pk_variant_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `fk_responsible_employee_id` - FK, 作为外键指向 `t_employees` 的主键.
- `fk_product_id`, `fk_variant_id` - FKs, 组合列作为外键指向 `t_product_variants` 的主键.
- `warehouse_change_quantity` - 仓库库存变化量.
- `shelf_change_quantity` - 货架库存变化量.

> `INDEX` 为索引定义，旨在优化数据库查询操作.

2.3. Details about Product Supply System

2.3.1. t_supply_records (供货记录)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_supply_records` (
  `pk_supply_record_id` INT NOT NULL,
  `fk_supply_product_id` INT NULL DEFAULT NULL,
  `fk_supply_variant_id` INT NULL DEFAULT NULL,
  `fk_supplier_id` INT NULL DEFAULT NULL,
  `supply_date` DATE NULL DEFAULT NULL,
  `supply_quantity` FLOAT NULL DEFAULT NULL,
  `total_price` FLOAT NULL DEFAULT NULL,
  `pay_term` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_supply_record_id`),
  INDEX `fk_supply_product_id` (`fk_supply_product_id` ASC, `fk_supply_variant_id` ASC) VISIBLE,
  INDEX `fk_supplier_id` (`fk_supplier_id` ASC) VISIBLE,
  CONSTRAINT `t_supply_records_ibfk_1`
    FOREIGN KEY (`fk_supply_product_id` , `fk_supply_variant_id`)
    REFERENCES `comprehensive_supermarket`.`t_product_variants` (`pk_product_id` , `pk_variant_id`),
  CONSTRAINT `t_supply_records_ibfk_2`
    FOREIGN KEY (`fk_supplier_id`)
    REFERENCES `comprehensive_supermarket`.`t_suppliers` (`pk_supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `fk_supply_product_id`, `fk_supply_variant_id` - FKs, 组合列作为外键指向 `t_product_variants` 的主键.
- `fk_supplier_id` - FK, 作为外键指向 `t_suppliers` 的主键.
- `pay_term` - 支付条例, 例如表示 "遵循3单位/10元购买……".

2.3.2. t_suppliers (供货商)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_suppliers` (
  `pk_supplier_id` INT NOT NULL,
  `company_name` VARCHAR(255) NULL DEFAULT NULL,
  `company_address` VARCHAR(255) NULL DEFAULT NULL,
  `company_website` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`pk_supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

2.3.3. t_supplier_contacts (供货商联系人)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_supplier_contacts` (
  `pk_contact_id` INT NOT NULL,
  `fk_supplier_id` INT NULL DEFAULT NULL,
  `contact_name` VARCHAR(255) NULL DEFAULT NULL,
  `contact_title` VARCHAR(255) NULL DEFAULT NULL,
  `phone_number` INT NULL DEFAULT NULL,
  `email` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`pk_contact_id`),
  INDEX `fk_supplier_id` (`fk_supplier_id` ASC) VISIBLE,
  CONSTRAINT `t_supplier_contacts_ibfk_1`
    FOREIGN KEY (`fk_supplier_id`)
    REFERENCES `comprehensive_supermarket`.`t_suppliers` (`pk_supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `fk_supplier_id` - FK, 作为外键指向 `t_suppliers` 的主键. 因为一家供货商可能有多个联系人.

2.4. Details about Product Sales System

2.4.1. t_transaction_records (交易记录)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_transaction_records` (
  `pk_transaction_id` INT NOT NULL,
  `fk_responsible_employee_id` INT NULL DEFAULT NULL,
  `fk_customer_id` INT NULL DEFAULT NULL,
  `transaction_date` DATE NULL DEFAULT NULL,
  `transaction_time` TIME NULL DEFAULT NULL,
  `transaction_way` VARCHAR(255) NULL DEFAULT NULL,
  `fk_promotion_id` INT NULL DEFAULT NULL,
  `initial_amount` FLOAT NULL DEFAULT NULL,
  `discounted_amount` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_transaction_id`),
  INDEX `fk_promotion_id` (`fk_promotion_id` ASC) VISIBLE,
  INDEX `fk_customer_id` (`fk_customer_id` ASC) VISIBLE,
  INDEX `fk_responsible_employee_id` (`fk_responsible_employee_id` ASC) VISIBLE,
  CONSTRAINT `transaction_records_ibfk_1`
    FOREIGN KEY (`fk_promotion_id`)
    REFERENCES `comprehensive_supermarket`.`t_reduction_promotions` (`pk_promotion_id`),
  CONSTRAINT `transaction_records_ibfk_2`
    FOREIGN KEY (`fk_customer_id`)
    REFERENCES `comprehensive_supermarket`.`t_customers` (`pk_customer_id`),
  CONSTRAINT `transaction_records_ibfk_3`
    FOREIGN KEY (`fk_responsible_employee_id`)
    REFERENCES `comprehensive_supermarket`.`t_employees` (`pk_employee_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `transaction_way` - 支付方式, 例如 "WeChat", "ApplePay".
- `fk_promotion_id` - FK, 满减促销活动的 id.
- `fk_customer_id` - FK, 消费的用户.
- `fk_responsible_employee_id` - FK, 该交易的负责员工.
- `initial_amount` - 在执行满减活动之前的总价.
- `discounted_amount` - 在执行满减活动之后的总价.

2.4.2. t_transaction_details (交易记录细节)

```
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_transaction_details` (
  `pk_transaction_id` INT NOT NULL,
  `pk_transaction_detail_id` INT NOT NULL,
  `fk_product_id` INT NULL DEFAULT NULL,
  `fk_variant_id` INT NULL DEFAULT NULL,
  `purchasing_quantity` FLOAT NULL DEFAULT NULL,
  `discounted_total_price` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_transaction_id`, `pk_transaction_detail_id`),
  INDEX `fk_product_id` (`fk_product_id` ASC, `fk_variant_id` ASC) VISIBLE,
  CONSTRAINT `transaction_details_ibfk_1`
    FOREIGN KEY (`pk_transaction_id`)
    REFERENCES `comprehensive_supermarket`.`t_transaction_records` (`pk_transaction_id`),
  CONSTRAINT `transaction_details_ibfk_2`
    FOREIGN KEY (`fk_product_id` , `fk_variant_id`)
    REFERENCES `comprehensive_supermarket`.`t_product_variants` (`pk_product_id` , `pk_variant_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `pk_transaction_id`, `pk_transaction_detail_id` - PKs, 同时 `pk_transaction_id` 作为外键指向 `t_transaction_records` 的主键. 因此, 该实体是一个弱实体. 每一条 "detail" 对于该交易中的每个 "product variant" 进行记录.
- `purchasing_quantity` - 产品购买数量.
- `discounted_total_price` - 产品本身打折后的价格. 因为产品的原始价格已经有记录, 可以直接查询, 因此没有在该表中记录.

> 超市的折扣有两种方式: 分产品的百分比折扣, 以及整体满减优惠. 在计算时, 先对于每个产品进行百分比折扣, 再在最后交易时进行满减优惠.

2.4.3. t_customers (顾客)

```
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_customers` (
  `pk_customer_id` INT NOT NULL,
  `customer_name` VARCHAR(255) NULL DEFAULT NULL,
  `phone_number` INT NULL DEFAULT NULL,
  `account_balance` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_customer_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

2.4.4. t_reduction_promotions (满减促销)

```
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_reduction_promotions` (
  `pk_promotion_id` INT NOT NULL,
  `threshold_amount` FLOAT NULL DEFAULT NULL,
  `discount_amount` FLOAT NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `end_date` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`pk_promotion_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

- `threshold_amount` - 满减优惠的触发阈值.
- `discount_amount` - 满减优惠的份额.

2.4.5. t_variant_discounts (产品变种折扣)

```
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_variant_discounts` (
  `pk_variant_discount_id` INT NOT NULL,
  `fk_discount_product_id` INT NULL DEFAULT NULL,
  `fk_discount_variant_id` INT NULL DEFAULT NULL,
  `variant_discount_rate` FLOAT NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
```

```sql
  `end_date` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`pk_variant_discount_id`),
  INDEX `fk_discount_product_id` (`fk_discount_product_id` ASC, `fk_discount_variant_id` ASC) VISIBLE,
  CONSTRAINT `t_variant_discounts_ibfk_1`
    FOREIGN KEY (`fk_discount_product_id` , `fk_discount_variant_id`)
    REFERENCES `comprehensive_supermarket`.`t_product_variants` (`pk_product_id` , `pk_variant_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

> 产品变种的折扣, 不同于满减优惠, 产品折扣是百分比的.

2.5. Details about Employee Management System

2.5.1. t_employees (职工表)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_employees` (
  `pk_employee_id` INT NOT NULL,
  `job_position` VARCHAR(255) NULL DEFAULT NULL,
  `salary` FLOAT NULL DEFAULT NULL,
  `phone_number` INT NULL DEFAULT NULL,
  `work_schedules` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_employee_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

2.6. Details about Marketing Reports System

2.6.1. t_sales reports (销售报告)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_sales_reports` (
  `pk_sales_report_id` INT NOT NULL,
  `report_type` VARCHAR(255) NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `end_date` DATE NULL DEFAULT NULL,
  `total_sales` FLOAT NULL DEFAULT NULL,
  `total_revenue` FLOAT NULL DEFAULT NULL,
  `create_time` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`pk_sales_report_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

2.6.2. t_inventory_reports (库存报告)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_inventory_reports` (
  `pk_inventory_report_id` INT NOT NULL,
  `report_type` VARCHAR(255) NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `end_date` DATE NULL DEFAULT NULL,
  `create_time` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`pk_inventory_report_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

2.6.3. t_expense_reports (开销报告)

```sql
CREATE TABLE IF NOT EXISTS `comprehensive_supermarket`.`t_expense_reports` (
  `pk_expense_report_id` INT NOT NULL,
  `report_type` VARCHAR(255) NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `end_date` DATE NULL DEFAULT NULL,
  `create_time` DATETIME NULL DEFAULT NULL,
  `supply_expense` FLOAT NULL DEFAULT NULL,
  `employee_expense` FLOAT NULL DEFAULT NULL,
  `other_expense` FLOAT NULL DEFAULT NULL,
  `total_expense` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`pk_expense_report_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```