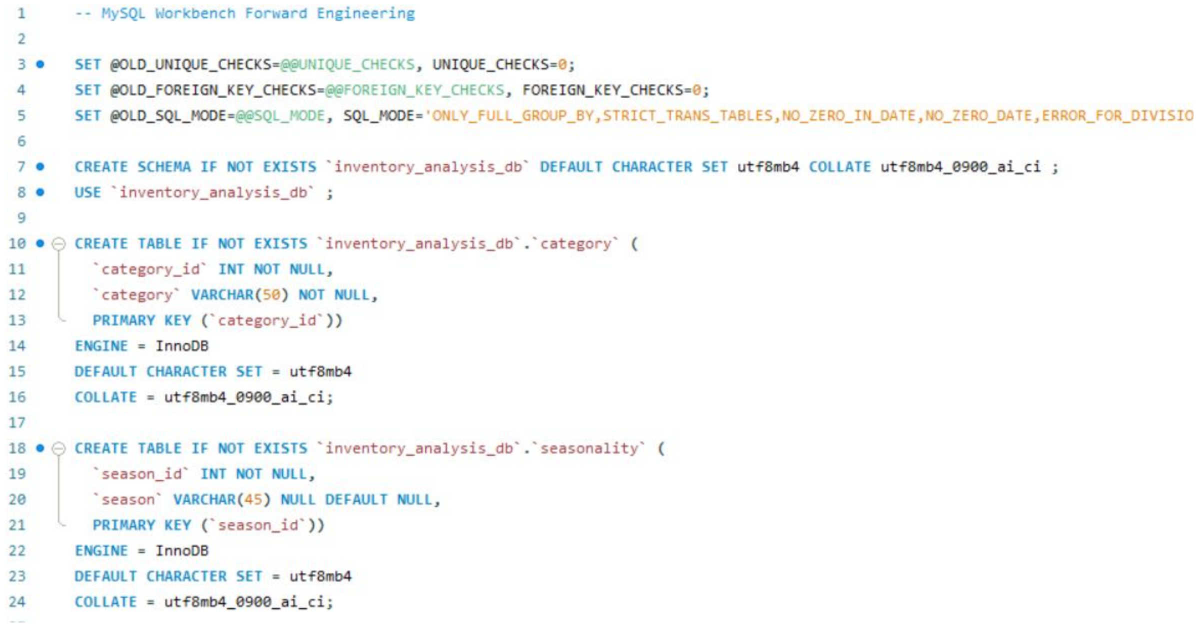
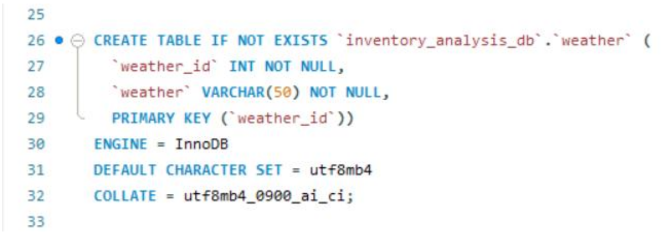
Solving Inventory Inefficiencies Using SQL

Vaibhav Pratap Singh (NIT Jamshedpur)

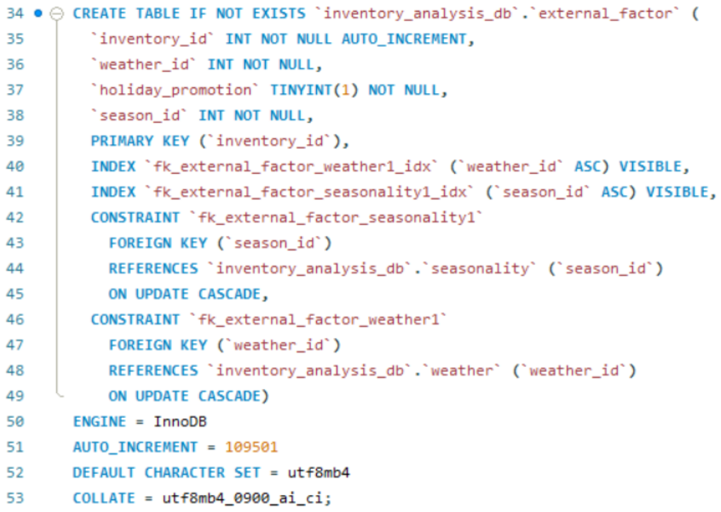
Prateeksha Singh (IIIT Lucknow)



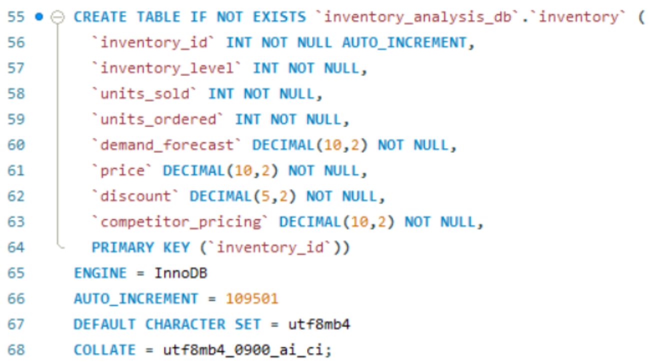
```sql
-- MySQL Workbench Forward Engineering

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_DIVISIO

CREATE SCHEMA IF NOT EXISTS `inventory_analysis_db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `inventory_analysis_db` ;

CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`category` (
  `category_id` INT NOT NULL,
  `category` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`category_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`seasonality` (
  `season_id` INT NOT NULL,
  `season` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`season_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```



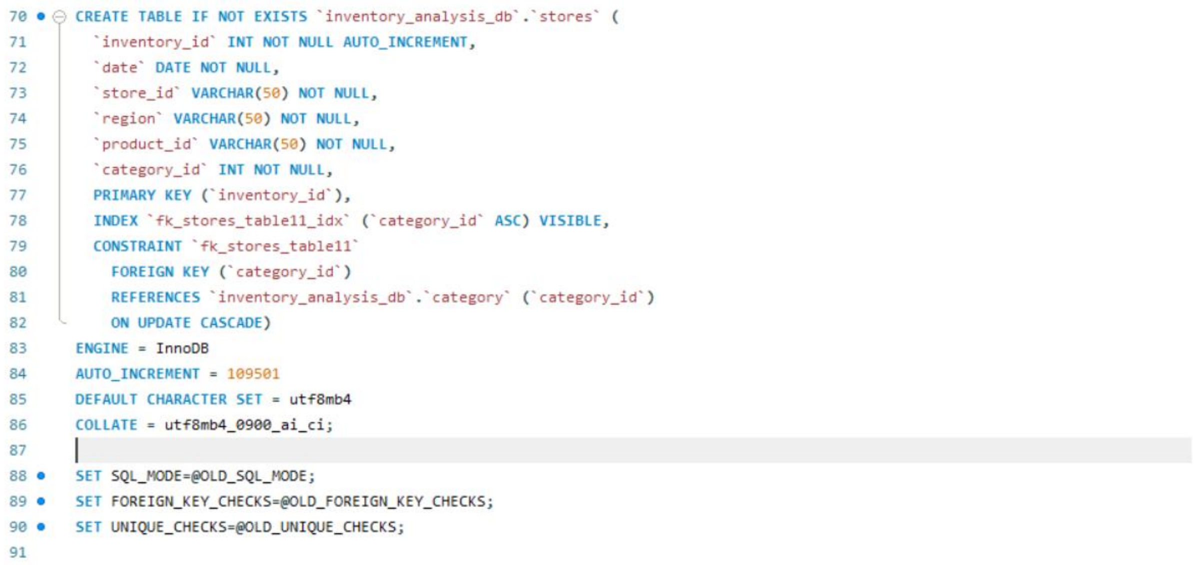
```sql

CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`weather` (
  `weather_id` INT NOT NULL,
  `weather` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`weather_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```



```sql
CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`external_factor` (
  `inventory_id` INT NOT NULL AUTO_INCREMENT,
  `weather_id` INT NOT NULL,
  `holiday_promotion` TINYINT(1) NOT NULL,
  `season_id` INT NOT NULL,
  PRIMARY KEY (`inventory_id`),
  INDEX `fk_external_factor_weather1_idx` (`weather_id` ASC) VISIBLE,
  INDEX `fk_external_factor_seasonality1_idx` (`season_id` ASC) VISIBLE,
  CONSTRAINT `fk_external_factor_seasonality1`
    FOREIGN KEY (`season_id`)
    REFERENCES `inventory_analysis_db`.`seasonality` (`season_id`)
    ON UPDATE CASCADE,
  CONSTRAINT `fk_external_factor_weather1`
    FOREIGN KEY (`weather_id`)
    REFERENCES `inventory_analysis_db`.`weather` (`weather_id`)
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 109501
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```



```sql
CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`inventory` (
  `inventory_id` INT NOT NULL AUTO_INCREMENT,
  `inventory_level` INT NOT NULL,
  `units_sold` INT NOT NULL,
  `units_ordered` INT NOT NULL,
  `demand_forecast` DECIMAL(10,2) NOT NULL,
  `price` DECIMAL(10,2) NOT NULL,
  `discount` DECIMAL(5,2) NOT NULL,
  `competitor_pricing` DECIMAL(10,2) NOT NULL,
  PRIMARY KEY (`inventory_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 109501
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```



```sql
CREATE TABLE IF NOT EXISTS `inventory_analysis_db`.`stores` (
  `inventory_id` INT NOT NULL AUTO_INCREMENT,
  `date` DATE NOT NULL,
  `store_id` VARCHAR(50) NOT NULL,
  `region` VARCHAR(50) NOT NULL,
  `product_id` VARCHAR(50) NOT NULL,
  `category_id` INT NOT NULL,
  PRIMARY KEY (`inventory_id`),
  INDEX `fk_stores_table11_idx` (`category_id` ASC) VISIBLE,
  CONSTRAINT `fk_stores_table11`
    FOREIGN KEY (`category_id`)
    REFERENCES `inventory_analysis_db`.`category` (`category_id`)
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 109501
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;
```

Database Name: inventory_analysis_db

Purpose: To analyze inventory performance using internal metrics and external factors such as weather, seasonality, and promotions.

Inventory ID Integration

Purpose	**Description**
inventory_id in Excel	Manually added as **surrogate key**.
Used In Tables	inventory, stores, external_factor
Benefits	Enables relational integrity, easy to join tables, and efficient SQL querying

Table: category

Column Name	Data Type	Description
category_id	INT (PK)	Unique ID for each category
category	VARCHAR(50)	Name of the product category

Table: seasonality

Column Name	Data Type	Description
season_id	INT (PK)	Unique ID for each season
season	VARCHAR(45)	Season name (e.g., Summer, Winter)

Table: weather

Column Name	Data Type	Description
weather_id	INT (PK)	Unique ID for each weather type
weather	VARCHAR(50)	Weather description (e.g., Rainy, Sunny)

Table: external_factor

Column Name	Data Type	Description
inventory_id	INT (PK, FK)	References inventory.inventory_id
weather_id	INT (FK)	References weather.weather_id
holiday_promotion	TINYINT(1)	1 = Holiday Promotion active, 0 = No promotion
season_id	INT (FK)	References seasonality.season_id

Table: inventory

Column Name	Data Type	Description
inventory_id	INT (PK)	Unique ID for inventory records
inventory_level	INT	Current stock level
units_sold	INT	Units sold in the given period
units_ordered	INT	Units ordered
demand_forecast	DECIMAL(10,2)	Predicted future demand
price	DECIMAL(10,2)	Selling price per unit

Column Name	Data Type	Description
discount	DECIMAL(5,2)	Discount applied on the product
competitor_pricing	DECIMAL(10,2)	Competitor’s price for comparison

Table: stores

Column Name	Data Type	Description
inventory_id	INT (PK)	References inventory.inventory_id (useful for aggregation and joining tables)
date	DATE	Date of inventory record
store_id	VARCHAR(50)	Unique store identifier
region	VARCHAR(50)	Store’s geographical region
product_id	VARCHAR(50)	Product code or SKU
category_id	INT (FK)	References category.category_id

Table Relationships:

1. stores.category_id → category.category_id
2. stores.inventory_id → inventory.inventory_id
3. external_factor.inventory_id → inventory.inventory_id
4. external_factor.weather_id → weather.weather_id
5. external_factor.season_id → seasonality.season_id