```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;


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;


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`

## Objective:

This database is designed to assess inventory performance by leveraging internal KPIs and incorporating external influences such as climate patterns, seasonal fluctuations, and holiday-based promotions.

---

## Inventory ID Implementation

- **Purpose:**
  A manually introduced surrogate key (`inventory_id`) is used in place of a natural key for better control over data integrity.

- **Tables Utilizing It:**
    - `inventory`
    - `stores`
    - `external_factor`
- **Advantages:**
  Promotes strong table linkages, simplifies join operations, and enhances query efficiency in SQL environments.

---

## Table Definitions

### 1. category

| Column | Data Type | Description |
|---|---|---|
| `category_id` | INT (Primary Key) | Unique identifier for each product category |
| `category` | VARCHAR(50) | Label or name of the product grouping |

---

### 2. seasonality

| Column | Data Type | Description |
|---|---|---|
| `season_id` | INT (Primary Key) | Distinct identifier for each season |
| `season` | VARCHAR(45) | Seasonal classification, e.g., Summer, Winter |

---

### 3. weather

| Column | Data Type | Description |
|---|---|---|
| `weather_id` | INT (Primary Key) | Unique ID for weather types |

| | | |
|---|---|---|
| weather | VARCHAR(50) | Description of the weather condition (e.g., Sunny, Rainy) |

---

**4. external_factor**

| Column | Data Type | Description |
|---|---|---|
| inventory_id | INT (Primary & Foreign Key) | Links to inventory.inventory_id |
| weather_id | INT (Foreign Key) | Connects to weather.weather_id |
| holiday_promotion | TINYINT(1) | Indicates if a promotion is active (1 = Yes, 0 = No) |
| season_id | INT (Foreign Key) | Connects to seasonality.season_id |

---

**5. inventory**

| Column | Data Type | Description |
|---|---|---|
| inventory_id | INT (Primary Key) | Unique identifier for inventory records |
| inventory_level | INT | Current quantity in stock |
| units_sold | INT | Number of units sold within a timeframe |
| units_ordered | INT | Quantity ordered for restocking |
| demand_forecast | DECIMAL(10,2) | Predicted demand for the future |
| price | DECIMAL(10,2) | Price per unit sold |
| discount | DECIMAL(5,2) | Discount applied to the item |
| competitor_pricing | DECIMAL(10,2) | Comparable pricing from competitors |

---

**6. stores**

| Column | Data Type | Description |
|---|---|---|

| | | |
|---|---|---|
| `inventory_id` | INT (Primary Key) | Connects to `inventory.inventory_id` for aggregations and joins |
| `date` | DATE | Date of the recorded inventory transaction |
| `store_id` | VARCHAR(50) | Unique identifier for each retail outlet |
| `region` | VARCHAR(50) | Geographical area where the store is located |
| `product_id` | VARCHAR(50) | Product SKU or item code |
| `category_id` | INT (Foreign Key) | Connects to `category.category_id` |

---

### Key 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`