**3. Please design three MySQL tables to store information about:**

**a. product: the “product” tables should include the name, category of the product; it**

**should also provide information to tell us when this product was added to the**

**table, and who added this product.**

**Solution:**

**Creating product table:**

CREATE TABLE product (

Product\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR (255) NOT NULL,

category VARCHAR (255) NOT NULL,

added\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

added\_by VARCHAR (255) NOT NULL

);

**Inserting values into product table:**

insert into product (name, category, added\_by)

values( 'iphone', 'electronics', 'Manju'),

('eggs', ' Grocery', ' sri'),

('Balloons','Decor', 'MS') ;

**Select \* from product;**

A screenshot of a computer

Description automatically generated

**b. product price: this table will store the current price information of each product**

**and we should be able to join it with the product table. It should include the price,**

**current discount percent(default to 0), the updated time and who updated it.**

**Solution:**

**Creating product\_price table:**

CREATE TABLE product\_price (

price\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_id INT,

price DECIMAL(10, 2) NOT NULL,

discount\_percent DECIMAL(5, 2) DEFAULT 0,

updated\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_by VARCHAR(100),

FOREIGN KEY (product\_id) REFERENCES product(product\_id)

);

**Inserting values into product\_price table:**

INSERT INTO product\_price (product\_id, price, discount\_percent, updated\_by)

VALUES

(1, 599.99, 10.00, 'Manju'),

(2, 39.99, 5.00, 'Sri'),

(3, 249.00, 0.00, 'ms'),

(3, 599.99, 10.00, 'Manju');

**Select \* from product\_price**:

A screenshot of a computer

Description automatically generated

**c. product price change log: this table will store the old and new value of the**

**“product price” table that’s impacted by any insert/update/delete and contains**

**information of who and when the operation was performed.**

**Solution:**

**Creating product\_price\_change\_table:**

CREATE TABLE product\_price\_change\_log (

log\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_id INT,

old\_price DECIMAL(10, 2),

new\_price DECIMAL(10, 2),

operation\_type ENUM('insert', 'update', 'delete'),

operation\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

operation\_by VARCHAR(100),

FOREIGN KEY (product\_id) REFERENCES product(product\_id)

);

**Inserting values into product\_price\_change\_log table:**

INSERT INTO product\_price\_change\_log (product\_id, old\_price, new\_price, operation\_type, operation\_by)

VALUES

(1, 599.99, 549.99, 'update', 'Manju'),

(2, 39.99, 37.99, 'update', 'Ms'),

(3, 249.00, 269.00, 'update', 'sri');

**Select \* from product\_price\_change\_log table:**

**A screenshot of a computer

Description automatically generated**

**Please list the table creation scripts in proper MySQL syntax for these three tables and a**

**query to join “product” table and “product price” table together to show the product name,**

**category, price, and who/when it gets updated.**

**Query to Join product and product price tables:**

SELECT

p.name,

p.category,

pp.price,

pp.updated\_by,

pp.updated\_time

FROM product p

JOIN product\_price pp ON p.product\_id = pp.product\_id;

**Result:**

**A screenshot of a computer

Description automatically generated**

**One-to-Many Relationship between "product" and "product\_price":**

Each product in the "product" table can have multiple price entries in the "product\_price" table.

Each price entry in the "product\_price" table belongs to only one product.

This is established through the product\_id foreign key in the "product\_price" table.

**One-to-Many Relationship between "product" and "product\_price\_change\_log":**

Each product in the "product" table can have multiple change log entries in the "product\_price\_change\_log" table.

Each change log entry in the "product\_price\_change\_log" table corresponds to only one product.

This is established through the product\_id foreign key in the "product\_price\_change\_log" table.

**Many-to-One Relationship between "product\_price" and "product":**

Each price entry in the "product\_price" table belongs to one product in the "product" table.

Each product in the "product" table can have multiple associated price entries in the "product\_price" table.

This is established through the product\_id foreign key in the "product\_price" table.

**Many-to-One Relationship between "product\_price\_change\_log" and "product":**

Each change log entry in the "product\_price\_change\_log" table corresponds to one product in the "product" table.

Each product in the "product" table can have multiple associated change log entries in the "product\_price\_change\_log" table.

This is established through the product\_id foreign key in the "product\_price\_change\_log" table.

These relationships illustrate how the data in the three tables are interconnected. The one-to-many relationships reflect that one entity (product) can have multiple related entities (price entries or change log entries), while the many-to-one relationships indicate that multiple related entities (price entries or change log entries) correspond to one entity (product). These relationships help maintain data integrity and organize the data effectively within a relational database system.