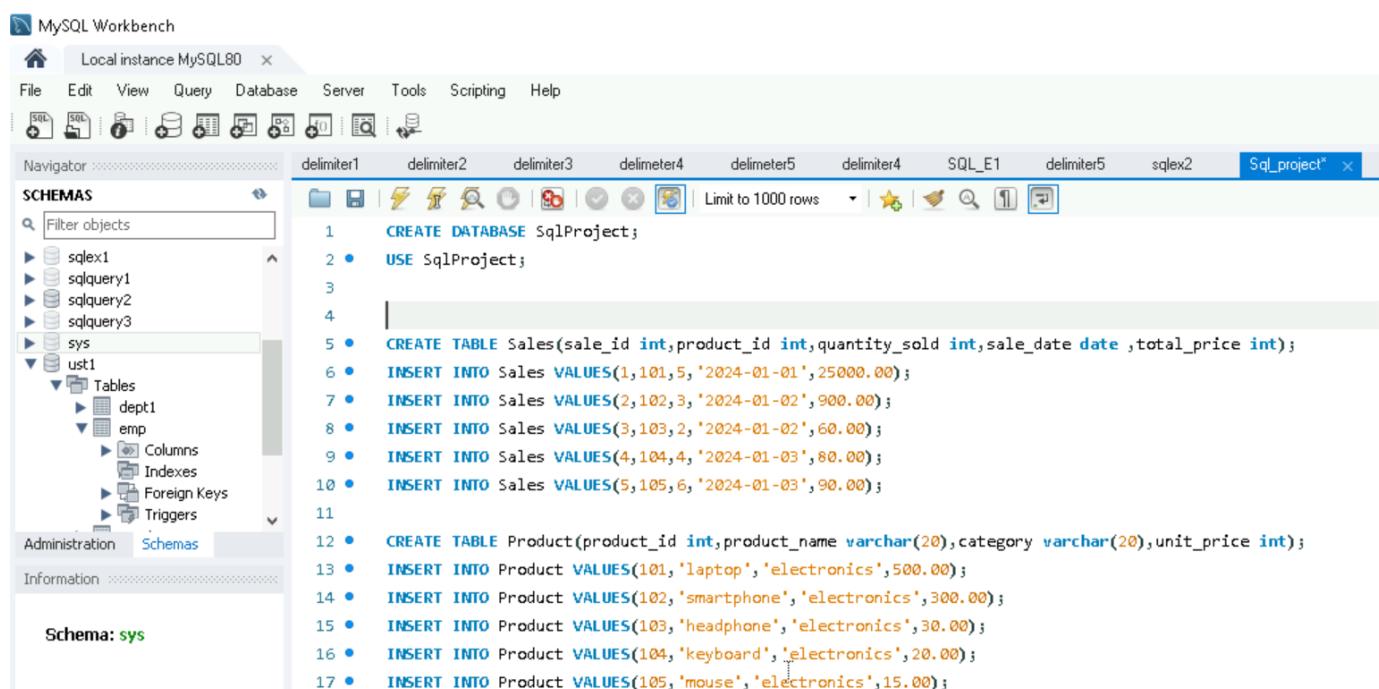


PROJECT 4

Project #4

- Q1. Find the product category with the highest average unit price.
- Q2. Identify products with total sales exceeding 30.
- Q3. Count the number of sales made in each month.
- Q4. Retrieve Sales Details for Products with 'Smart' in Their Name
- Q5. Determine the average quantity sold for products with a unit price greater than \$100.

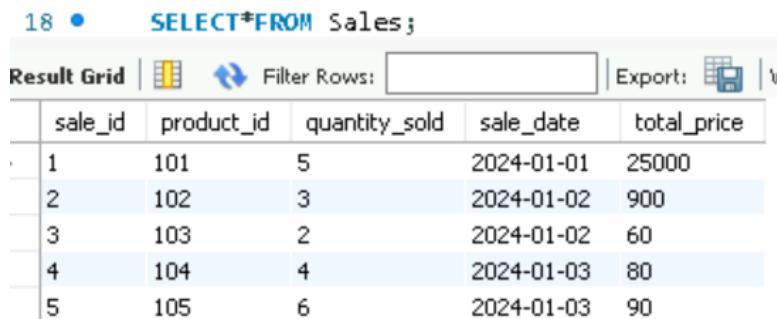
CREATE DATABASE, CREATE TABLE AND INSERT VALUES:



The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' tab, there is a tree view of databases: sqlex1, sqlex2, sqlex3, sys, and ust1. Under 'ust1', there is a 'Tables' node which contains 'dept1' and 'emp'. The 'emp' table has four child nodes: 'Columns', 'Indexes', 'Foreign Keys', and 'Triggers'. The main SQL editor window contains the following SQL code:

```
1 • CREATE DATABASE SqlProject;
2 • USE SqlProject;
3
4
5 • CREATE TABLE Sales(sale_id int,product_id int,quantity_sold int,sale_date date ,total_price int);
6 • INSERT INTO Sales VALUES(1,101,5,'2024-01-01',25000.00);
7 • INSERT INTO Sales VALUES(2,102,3,'2024-01-02',900.00);
8 • INSERT INTO Sales VALUES(3,103,2,'2024-01-02',60.00);
9 • INSERT INTO Sales VALUES(4,104,4,'2024-01-03',80.00);
10 • INSERT INTO Sales VALUES(5,105,6,'2024-01-03',90.00);
11
12 • CREATE TABLE Product(product_id int,product_name varchar(20),category varchar(20),unit_price int);
13 • INSERT INTO Product VALUES(101,'laptop','electronics',500.00);
14 • INSERT INTO Product VALUES(102,'smartphone','electronics',300.00);
15 • INSERT INTO Product VALUES(103,'headphone','electronics',30.00);
16 • INSERT INTO Product VALUES(104,'keyboard','electronics',20.00);
17 • INSERT INTO Product VALUES(105,'mouse','electronics',15.00);
```

Sales Table:



The screenshot shows the results of a SELECT query on the 'Sales' table. The results are displayed in a grid format with the following columns: sale_id, product_id, quantity_sold, sale_date, and total_price. The data is as follows:

sale_id	product_id	quantity_sold	sale_date	total_price
1	101	5	2024-01-01	25000
2	102	3	2024-01-02	900
3	103	2	2024-01-02	60
4	104	4	2024-01-03	80
5	105	6	2024-01-03	90

PROJECT 4

Product Table:

```
22 •     SELECT * FROM Product;
```

	product_id	product_name	category	unit_price
▶	101	laptop	electronics	500
	102	smartphone	electronics	300
	103	headphone	electronics	30
	104	keyboard	electronics	20
	105	mouse	electronics	15

1.QUERY:

```
23 •     SELECT category, AVG(unit_price) AS avg_price FROM Product GROUP BY category ORDER BY avg_price DESC LIMIT 1;
```

OUTPUT:

	category	avg_price
▶	electronics	173.0000

2.QUERY:

```
24 •     SELECT product_name FROM Product WHERE unit_price NOT IN (SELECT unit_price FROM Product WHERE unit_price=30);
```

OUTPUT:

	product_name
▶	laptop
	smartphone
	keyboard
	mouse

3.QUERY:

```
25 •     SELECT MONTH(sale_date) AS sale_month, COUNT(*) AS total_sales FROM Sales GROUP BY MONTH(sale_date) ORDER BY sale_month;
```

OUTPUT:

PROJECT 4

Result Grid | Filter Rows

	sale_month	total_sales
▶	1	5

4.QUERY:

```
29 •   SELECT * FROM Sales WHERE product_id =(SELECT product_id FROM Product WHERE product_name like 'smart%');  
--
```

OUTPUT:

Result Grid | Filter Rows: | Export: |

	sale_id	product_id	quantity_sold	sale_date	total_price
▶	2	102	3	2024-01-02	900

5.QUERY:

```
32 •   SELECT AVG(quantity_sold) as avg_quantity_sold FROM Sales WHERE product_id in (SELECT product_id FROM Product WHERE unit_price >100);
```

OUTPUT:

Result Grid | Filter Rows |

	avg_quantity_sold
▶	4.0000