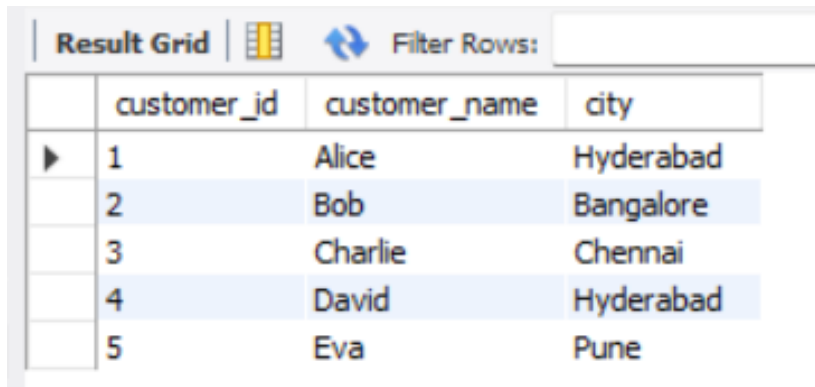


Task – 3 Out Puts

- **Basic select**

```
SELECT *FROM customers_task_3;
```

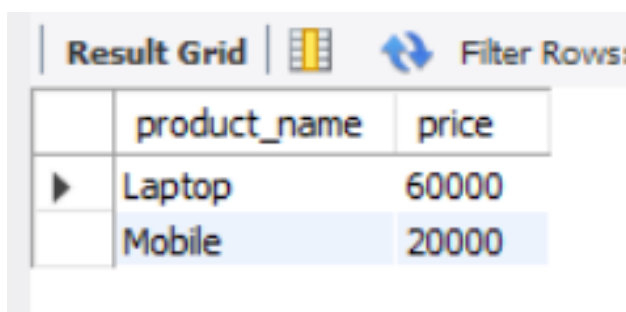


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with 4 columns: an empty header cell, 'customer_id', 'customer_name', and 'city'. There are 5 data rows. The first row is highlighted with a play button icon in the first column. The second and fourth rows are highlighted with a blue background.

	customer_id	customer_name	city
▶	1	Alice	Hyderabad
	2	Bob	Bangalore
	3	Charlie	Chennai
	4	David	Hyderabad
	5	Eva	Pune

- **where (Filtering)**

```
SELECT product_name, price  
FROM products_task_3  
WHERE price > 10000;
```

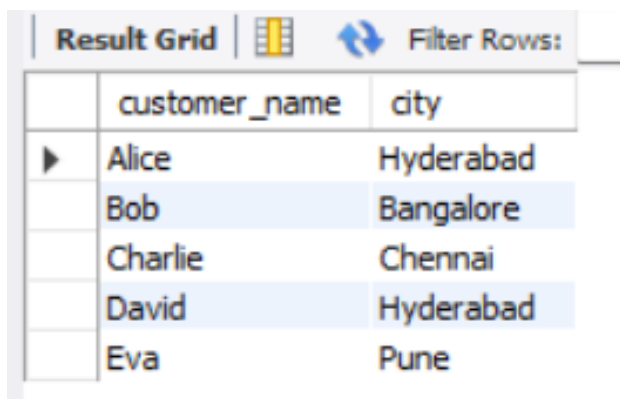


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with 3 columns: an empty header cell, 'product_name', and 'price'. There are 2 data rows. The first row is highlighted with a play button icon in the first column. The second row is highlighted with a blue background.

	product_name	price
▶	Laptop	60000
	Mobile	20000

- **Order by**

```
SELECT customer_name, city  
FROM customers_task_3  
ORDER BY customer_name ASC;
```



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with 3 columns: an empty header cell, 'customer_name', and 'city'. There are 5 data rows. The first row is highlighted with a play button icon in the first column. The second, fourth, and fifth rows are highlighted with a blue background.

	customer_name	city
▶	Alice	Hyderabad
	Bob	Bangalore
	Charlie	Chennai
	David	Hyderabad
	Eva	Pune

- **GROUP BY with Aggregates**

```
SELECT c.customer_name, SUM(p.price * o.quantity) AS total_spent
FROM orders_task_3 o
INNER JOIN customers_task_3 c ON o.customer_id = c.customer_id
INNER JOIN products_task_3 p ON o.product_id = p.product_id
GROUP BY c.customer_name
ORDER BY total_spent DESC;
```

Result Grid			Filter Rows:
	customer_name	total_spent	
▶	Alice	63000	
	Eva	60000	
	Bob	52000	
	Charlie	4500	
	David	1600	

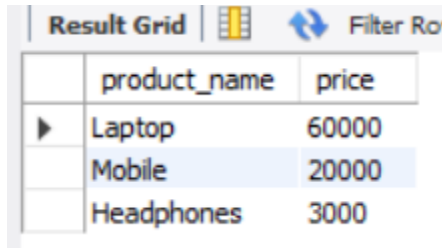
- **JOIN**

```
SELECT o.order_id, c.customer_name, p.product_name, o.quantity, o.order_date
FROM orders_task_3 o
INNER JOIN customers_task_3 c ON o.customer_id = c.customer_id
INNER JOIN products_task_3 p ON o.product_id = p.product_id;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content
	order_id	customer_name	product_name	quantity	order_date
▶	1006	Eva	Laptop	1	2023-03-20 00:00:00
	1001	Alice	Laptop	1	2023-01-10 00:00:00
	1002	Bob	Mobile	2	2023-01-15 00:00:00
	1007	Bob	Headphones	4	2023-04-01 00:00:00
	1003	Alice	Headphones	1	2023-02-01 00:00:00
	1004	Charlie	Keyboard	3	2023-02-05 00:00:00
	1005	David	Mouse	2	2023-03-12 00:00:00

- **LIMIT**

```
SELECT product_name, price  
FROM products_task_3  
ORDER BY price DESC  
LIMIT 3;
```

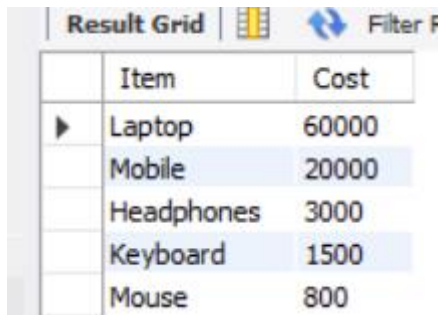


The screenshot shows a 'Result Grid' window with a table containing three rows. The first row is 'Laptop' with a price of 60000. The second row is 'Mobile' with a price of 20000. The third row is 'Headphones' with a price of 3000. The table has columns 'product_name' and 'price'.

	product_name	price
▶	Laptop	60000
	Mobile	20000
	Headphones	3000

- **Using Aliases (AS)**

```
SELECT p.product_name AS Item, p.price AS Cost  
FROM products_task_3 p;
```

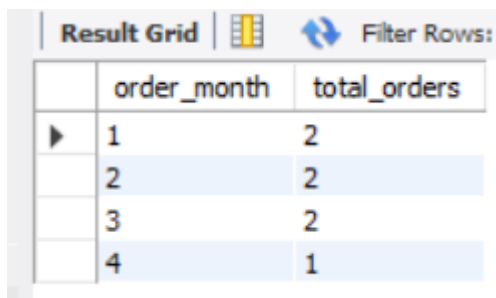


The screenshot shows a 'Result Grid' window with a table containing five rows. The first row is 'Laptop' with a cost of 60000. The second row is 'Mobile' with a cost of 20000. The third row is 'Headphones' with a cost of 3000. The fourth row is 'Keyboard' with a cost of 1500. The fifth row is 'Mouse' with a cost of 800. The table has columns 'Item' and 'Cost'.

	Item	Cost
▶	Laptop	60000
	Mobile	20000
	Headphones	3000
	Keyboard	1500
	Mouse	800

- **Date Functions**

```
SELECT MONTH(order_date) AS order_month, COUNT(*) AS total_orders  
FROM orders_task_3  
GROUP BY order_month;
```



The screenshot shows a 'Result Grid' window with a table containing four rows. The first row is '1' with a total of 2 orders. The second row is '2' with a total of 2 orders. The third row is '3' with a total of 2 orders. The fourth row is '4' with a total of 1 order. The table has columns 'order_month' and 'total_orders'.

	order_month	total_orders
▶	1	2
	2	2
	3	2
	4	1