

Create the Orders Table

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view with 'sales' selected. The main editor shows the following SQL code:

```
1 • CREATE DATABASE Sales;
2 • Use sales;
3 • CREATE TABLE Orders (
4     Order_Id INT PRIMARY KEY UNIQUE NOT NULL,
5     Customer_name VARCHAR(255) NOT NULL,
6     Product_Category VARCHAR(100) NOT NULL,
7     Ordered_item VARCHAR(255) NOT NULL,
8     Contact_No VARCHAR(15) NOT NULL
9 );
10 • select * from orders;
```

Add a New Column

The screenshot shows the same database management tool interface. The SQL code in the main editor is updated to include adding a new column:

```
1 • CREATE DATABASE Sales;
2 • Use sales;
3 • CREATE TABLE Orders (
4     Order_Id INT PRIMARY KEY UNIQUE NOT NULL,
5     Customer_name VARCHAR(255) NOT NULL,
6     Product_Category VARCHAR(100) NOT NULL,
7     Ordered_item VARCHAR(255) NOT NULL,
8     Contact_No VARCHAR(15) NOT NULL
9 );
10 • select * from orders;
11 • ALTER TABLE Orders
12     ADD order_quantity INT NOT NULL;
13 • select * from orders;
14
```

Below the SQL editor, the 'Result Grid' is visible, showing the columns of the 'Orders' table:

Order_Id	Customer_name	Product_Category	Ordered_item	Contact_No	order_quantity
NULL	NULL	NULL	NULL	NULL	NULL

Rename the Orders Table

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' tree is expanded to show the 'sales' database, which contains a table named 'sales_orders'. The main editor displays the following SQL script:

```
1 • CREATE DATABASE Sales;
2 • Use sales;
3 • CREATE TABLE Orders (
4     Order_Id INT PRIMARY KEY UNIQUE NOT NULL,
5     Customer_name VARCHAR(255) NOT NULL,
6     Product_Category VARCHAR(100) NOT NULL,
7     Ordered_item VARCHAR(255) NOT NULL,
8     Contact_No VARCHAR(15) NOT NULL
9 );
10 • select * from orders;
11 • ALTER TABLE Orders
12     ADD order_quantity INT NOT NULL;
13 • select * from orders;
14 • ALTER TABLE Orders
15     RENAME TO sales_orders;
16
```

Insert 10 Rows into the sales_orders Table

The screenshot shows the same SQL IDE interface. The SQL script in the editor now includes an INSERT statement to add 10 rows of data into the 'sales_orders' table:

```
14 • ALTER TABLE Orders
15     RENAME TO sales_orders;
16 • INSERT INTO sales_orders (Order_Id, Customer_name, Product_Category, Ordered_item, Contact_No, order_quantity) VALUES
17     (1, 'Raj', 'Electronics', 'Smartphone', '123-456-7890', 2),
18     (2, 'Amit', 'Clothing', 'Jacket', '234-567-8901', 1),
19     (3, 'Priya', 'Home', 'Blender', '345-678-9012', 1),
20     (4, 'Nina', 'Electronics', 'Laptop', '456-789-0123', 1),
21     (5, 'Sita', 'Kitchen', 'Cookware Set', '567-890-1234', 3),
22     (6, 'Rahul', 'Sports', 'Tennis Racket', '678-901-2345', 1),
23     (7, 'Asha', 'Books', 'Novel', '789-012-3456', 4),
24     (8, 'Vikram', 'Beauty', 'Perfume', '890-123-4567', 1),
25     (9, 'Neha', 'Electronics', 'Headphones', '901-234-5678', 2),
26     (10, 'Karan', 'Clothing', 'T-Shirt', '012-345-6789', 5);
27 • select * from sales_orders;
```

Below the script, the 'Result Grid' shows the first 5 rows of the data inserted:

Order_Id	Customer_name	Product_Category	Ordered_item	Contact_No	order_quantity
1	Raj	Electronics	Smartphone	123-456-7890	2
2	Amit	Clothing	Jacket	234-567-8901	1
3	Priya	Home	Blender	345-678-9012	1
4	Nina	Electronics	Laptop	456-789-0123	1
5	Sita	Kitchen	Cookware Set	567-890-1234	3

Retrieve customer_name and Ordered_Item

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'football', 'football2', 'institute', and 'sales' schemas. The 'sales' schema is selected, showing 'sales_orders' as a table. The main editor displays the following SQL code:

```
16 • INSERT INTO sales_orders (Order_Id, Customer_name, Product_Category, Ordered_item, Contact_No, order_quantity)
17   (1, 'Raj', 'Electronics', 'Smartphone', '123-456-7890', 2),
18   (2, 'Amit', 'Clothing', 'Jacket', '234-567-8901', 1),
19   (3, 'Priya', 'Home', 'Blender', '345-678-9012', 1),
20   (4, 'Nina', 'Electronics', 'Laptop', '456-789-0123', 1),
21   (5, 'Sita', 'Kitchen', 'Cookware Set', '567-890-1234', 3),
22   (6, 'Rahul', 'Sports', 'Tennis Racket', '678-901-2345', 1),
23   (7, 'Asha', 'Books', 'Novel', '789-012-3456', 4),
24   (8, 'Vikram', 'Beauty', 'Perfume', '890-123-4567', 1),
25   (9, 'Neha', 'Electronics', 'Headphones', '901-234-5678', 2),
26   (10, 'Karan', 'Clothing', 'T-Shirt', '012-345-6789', 5);
27 • select * from sales_orders;
28 • SELECT Customer_name, Ordered_item FROM sales_orders;
29
```

Below the SQL editor, the 'Result Grid' shows the output of the query. It has columns 'Customer_name' and 'Ordered_item'.

Customer_name	Ordered_item
Raj	Smartphone
Amit	Jacket
Priya	Blender
Nina	Laptop
Sita	Cookware Set

Update the Product for Any Row

The screenshot shows the same database management tool interface. The SQL editor now contains the following code:

```
19   (3, 'Priya', 'Home', 'Blender', '345-678-9012', 1),
20   (4, 'Nina', 'Electronics', 'Laptop', '456-789-0123', 1),
21   (5, 'Sita', 'Kitchen', 'Cookware Set', '567-890-1234', 3),
22   (6, 'Rahul', 'Sports', 'Tennis Racket', '678-901-2345', 1),
23   (7, 'Asha', 'Books', 'Novel', '789-012-3456', 4),
24   (8, 'Vikram', 'Beauty', 'Perfume', '890-123-4567', 1),
25   (9, 'Neha', 'Electronics', 'Headphones', '901-234-5678', 2),
26   (10, 'Karan', 'Clothing', 'T-Shirt', '012-345-6789', 5);
27 • select * from sales_orders;
28 • SELECT Customer_name, Ordered_item FROM sales_orders;
29 • UPDATE sales_orders
30   SET Ordered_item = 'Smartwatch'
31   WHERE Order_Id = 1;
32 • select * from sales_orders;
--
```

The 'Result Grid' now displays the updated data with columns 'Order_Id', 'Customer_name', 'Product_Category', 'Ordered_item', 'Contact_No', and 'order_quantity'.

Order_Id	Customer_name	Product_Category	Ordered_item	Contact_No	order_quantity
1	Raj	Electronics	Smartwatch	123-456-7890	2
2	Amit	Clothing	Jacket	234-567-8901	1
3	Priya	Home	Blender	345-678-9012	1
4	Nina	Electronics	Laptop	456-789-0123	1
5	Sita	Kitchen	Cookware Set	567-890-1234	3

Delete the sales_orders Table

The screenshot shows a database management interface with a left-hand sidebar and a main SQL editor window.

Left Sidebar (SCHEMAS):

- Filter objects
- football
- football2
- institute
 - Tables
 - football_players2
 - managers
 - Views
 - Stored Procedures
 - Functions
- sales**
 - Tables
 - Views
 - Stored Procedures
 - Functions

Main SQL Editor:

Limit to 1000 rows

```
14 • ALTER TABLE Orders
15 RENAME TO sales_orders;
16 • INSERT INTO sales_orders (Order_Id, Customer_name, Product_Category, Ordered_item, Contact_No, order_quantity) VALUES
17 (1, 'Raj', 'Electronics', 'Smartphone', '123-456-7890', 2),
18 (2, 'Amit', 'Clothing', 'Jacket', '234-567-8901', 1),
19 (3, 'Priya', 'Home', 'Blender', '345-678-9012', 1),
20 (4, 'Nina', 'Electronics', 'Laptop', '456-789-0123', 1),
21 (5, 'Sita', 'Kitchen', 'Cookware Set', '567-890-1234', 3),
22 (6, 'Rahul', 'Sports', 'Tennis Racket', '678-901-2345', 1),
23 (7, 'Asha', 'Books', 'Novel', '789-012-3456', 4),
24 (8, 'Vikram', 'Beauty', 'Perfume', '890-123-4567', 1),
25 (9, 'Neha', 'Electronics', 'Headphones', '901-234-5678', 2),
26 (10, 'Karan', 'Clothing', 'T-Shirt', '012-345-6789', 5);
27 • select * from sales_orders;
28 • SELECT Customer_name, Ordered_item FROM sales_orders;
29 • UPDATE sales_orders
30 SET Ordered_item = 'Smartwatch'
31 WHERE Order_Id = 1;
32 • select * from sales_orders;
33 • DROP TABLE sales_orders;
34
```

Administration Schemas
Information

Schema: sales