

\* C - Create  
R - Read  
U - Update  
D - Destroy

# Create

\* CREATE TABLE products (

    id INT NOT NULL,  
    name STRING,  
    price MONEY,  
    PRIMARY KEY (id) // will be unique for every product  
)

INSERT INTO products  
VALUES (1, "Pen", 1.20)

\* INSERT INTO products (id, name)  
VALUES (1, "Pencil")

\* INSERT INTO products (name, price)  
VALUES ("Rubber", 2) will not work as id NULL

## #Read

```
SELECT * FROM 'products';
```

↳ means select everything from table to show up

```
SELECT name, price FROM 'products';
```

will just display name and price

\* where : to select specific data

```
SELECT column1, column2, ...  
FROM table name  
WHERE condition;
```

```
SELECT * FROM 'products' WHERE id=1;  
// will show single row
```

## #Update

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

\* UPDATE products will set the price of all products to  
SET price = 0.80 0.8

\* UPDATE products  
SET price = 0.80  
WHERE id = 2

\* To update the full Table

ALTER TABLE table name

ADD column\_name datatype;

\* ALTER TABLE products  
ADD stock INT

i	id	name	price	stock
1	1	Pen	1.2	Null
2	2	Pencil	0.8	Null

Products

i	id 	name	price	stock
1	1	Pen	1.20	32
2	2	Pencil	0.80	12

UPDATE products  
SET stock = 32  
WHERE id = 1

UPDATE products  
SET stock = 12  
WHERE id = 2

## #DELETE

```
DELETE FROM table_name  
WHERE condition;
```

\* DELETE FROM products

WHERE id = 2 // if this statement is not there it will delete all of the table

\* INSERT INTO products  
VALUES (2, "pencil", 0.8)

\* CREATE TABLE orders (

id INT NOT NULL,

order\_number INT,

customer\_id INT, // foreign key

product\_id INT,

PRIMARY KEY (id), primary key means it accepts unique values

FOREIGN KEY (customer\_id) REFERENCES customers (id),

name of the table ]

name of the key that will be primary key in the table ↴  
FOREIGN KEY (product\_id) REFERENCES products (id),

INSERT INTO orders

VALUES (1, 4362, 2, 1)

i	id	order_number	customer_id	product_id
1		4362	2	1

~~SELECT column\_name(s)  
FROM table1  
INNER JOIN table2 ON table1.column\_name = table2.column\_name;~~

## Syntax to match foreign key

- SELECT orders.order\_number, customers.first\_name, customers.last\_name, customers.address //all the fields we want to join in our new table
- FROM orders <sup>from this table there will be a foreign key match</sup>
- INNER JOIN customers ON orders.customer\_id = <sup>foreign key</sup> customers.id <sup>we want to join the fields which will match with our primary key</sup>

The screenshot shows the MySQL Workbench environment. On the left, there's a sidebar with tabs for Table, View, Index, Trigger, and Syntax. Under Table, three tables are listed: customers, orders, and products. The orders table is currently selected. In the center, a query editor window displays the following SQL code:

```
1 SELECT orders.order_number, customers.first_name, customers.last_name, customers.address
2 FROM orders
3 INNER JOIN customers ON orders.customer_id = customers.id
```

Below the query editor is a results grid. It has four columns: order\_number, first\_name, last\_name, and address. A single row is visible, corresponding to the query results:

order_number	first_name	last_name	address
4362	Angela	Yu	12 Sunset Drive

## Final Output

id	order_number	customer_id	product_id
1	4362	2	1

## orders table

```
CREATE TABLE orders (
  id INT NOT NULL,
  order_number INT,
  customer_id INT, //foreign key
  product_id INT,
  PRIMARY KEY (id), primary key means it accepts unique values
  FOREIGN KEY (customer_id) REFERENCES customers (id), name of the table
  FOREIGN KEY (product_id) REFERENCES products (id); name of the key that will be primary key in the table
```

## \* Join order with the products

```
CREATE TABLE orders (
    id INT NOT NULL,
    order-number INT,
    customer_id INT, //foreign key
    product_id INT,
    PRIMARY KEY (id),
    FOREIGN KEY (customer_id) REFERENCES customers (id),
    FOREIGN KEY (product_id) REFERENCES products (id);
```

name of the key that will be primary key in the table

name of the table

```
SELECT order.order-number, products.name, products.price,
       products.stock
  FROM order
 INNER JOIN products ON orders.product_id =
          product.id
```