

Newton School | Krish Vigyan Kendri | ML-powered payroll | Home | payroll task | GitHub Dashboard

← → ⌂ my.newtonschool.co/playground/database/pglh85ss25zb

Finish update

Back | MySQL

KV ⚙ Run

MySQL

```
25 (205, 6, '2024-01-28', 1200.00);
26
27 adata retrieval
28 SELECT * FROM orders1;
29 SELECT * FROM payments;
...  
...
```

OUTPUT

ERROR

order_id	customer_id	order_date	total_amount
1	101	2024-01-10	2500.00
2	102	2024-01-12	1800.00
3	103	2024-01-15	3200.00
4	101	2024-01-20	1500.00
5	104	2024-01-25	4000.00
payment_id	order_id	payment_date	amount
201	1	2024-01-11	2500.00
202	2	2024-01-13	1800.00
203	3	2024-01-16	3000.00
204	5	2024-01-26	4000.00
205	6	2024-01-28	1200.00

my.newtonschool.co/playground/databases/pgjno3ss2cjd

Back | MySQL 

KV Run Save

MySQL ▾

```
48 #BETWEEN
49 SELECT * FROM orders1
50 WHERE total_amount BETWEEN 2000 AND 4000;
51 #IN / NOT IN
52 SELECT * FROM orders1
53 WHERE customer_id IN (101, 103);
54 SELECT * FROM orders1
55 WHERE customer_id NOT IN (102);
56
```

OUTPUT ERROR

order_id	customer_id	order_date	total_amount
1	101	2024-01-10	2500.00
3	103	2024-01-15	3200.00
5	104	2024-01-25	4000.00

order_id	customer_id	order_date	total_amount
1	101	2024-01-10	2500.00
3	103	2024-01-15	3200.00
4	101	2024-01-20	1500.00

order_id	customer_id	order_date	total_amount
1	101	2024-01-10	2500.00
3	103	2024-01-15	3200.00

News for you

07:52 PM

Newton School Krishi Vigyan Kendra ML powered payroll Home payroll task GitHub Dashboard

← → ⌂ my.newtonschool.co/playground/database/pglh85ss25zb

Back MySQL KV Run Save

MySQL

```
44 SELECT * FROM orders1 ORDER BY total_amount ASC;
45 SELECT * FROM orders1 ORDER BY order_date DESC;
```

OUTPUT ERROR

order_id	customer_id	order_date	total_amount
4	101	2024-01-20	1500.00
2	102	2024-01-12	1800.00
1	101	2024-01-10	2500.00
3	103	2024-01-15	3200.00
5	104	2024-01-25	4000.00

order_id	customer_id	order_date	total_amount
5	104	2024-01-25	4000.00
4	101	2024-01-20	1500.00
3	103	2024-01-15	3200.00
2	102	2024-01-12	1800.00
1	101	2024-01-10	2500.00

← → ⌂ my.newtonschool.co/playground/database/pglh85ss25zb

Finish update

Back | MySQL

X Run Save

MySQL

```
31 WHERE condition
32 SELECT * FROM orders1 WHERE total_amount > 2000;
33 Aggregate functions
34 SELECT SUM(total_amount) AS total_sales FROM orders1;
35 SELECT AVG(total_amount) AS avg_order_value FROM orders1;
36 SELECT MAX(total_amount) FROM orders1;
37 SELECT MIN(total_amount) FROM orders1;
38 SELECT COUNT(*) AS total_orders FROM orders1;
39
```

OUTPUT ERROR

order_id	customer_id	order_date	total_amount
1	101	2024-01-10	2500.00
3	103	2024-01-15	3200.00
5	104	2024-01-25	4000.00

total_sales

13000.00

avg.order.value

2600.000000

MAX(total.amount)

4000.00

MIN(total.amount)

29% ENG 02:18 PM

← Back | MySQL ⚡

KV

@

▶ Run

Save

MySQL ▾

```
93  #VIEW
94  CREATE VIEW order_payment_view AS
95  SELECT o.order_id, o.customer_id, o.total_amount, p.payment_date, p.amount
96  FROM orders o
97  LEFT JOIN payments p
98  ON o.order_id = p.order_id;
99
100 SELECT * FROM order_payment_view;
```

OUTPUT

ERROR

order_id	customer_id	total_amount	payment_date	amount
1	101	2500.00	2024-01-11	2500.00
2	102	1800.00	2024-01-13	1800.00
3	103	3200.00	2024-01-16	3000.00
4	101	1500.00	NULL	NULL
5	104	4000.00	2024-01-26	4000.00