

Online SQL Editor

programiz.com/sql/online-compiler/

Guest

Programiz

Online SQL Editor

exness

625,000+ traders choose Exness every month

Get started

Interactive SQL Course

ACCIDENT [-]

report\_number [int]

accd\_date [date]

location [varchar(100)]

CAR [-]

regno [int]

model [varchar(100)]

year [int]

Customers [-]

customer\_id [int]

first\_name [varchar(100)]

last\_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order\_id [integer]

item [varchar(100)]

amount [integer]

customer\_id [integer]

OWNS [-]

driver\_id [int]

regno [int]

PARTICIPATED [-]

driver\_id [int]

regno [int]

Input

```
-- Create PERSON table
CREATE TABLE PERSON (
  driver_id INT PRIMARY KEY,
  name VARCHAR(100),
  address VARCHAR(100)
);

-- Create CAR table
CREATE TABLE CAR (
  regno INT PRIMARY KEY,
  model VARCHAR(100),
  year INT
);

-- Create ACCIDENT table
CREATE TABLE ACCIDENT (
  report_number INT PRIMARY KEY,
  accd_date DATE,
  location VARCHAR(100)
);

-- Create OWNS table with foreign keys
```

Run SQL

Available Tables

Output

SQL query successfully executed. However, the result set is empty.

ACCIDENT

report_number	accd_date	location
10	2023-01-05	Intersection A
11	2023-02-10	Highway B
12	2023-03-15	Street C
13	2023-04-20	Road D
14	2023-05-25	Avenue E

CAR

regno	model	year
101	Toyota Camry	2008
102	Honda Civic	2008
103	Ford Mustang	2009
104	Chevrolet Malibu	2010
105	Nissan Altima	2011

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders

Input

```
-- Create OWNS table with foreign keys
CREATE TABLE OWNS (
  driver_id INT,
  regno INT,
  FOREIGN KEY (driver_id) REFERENCES PERSON(driver_id),
  FOREIGN KEY (regno) REFERENCES CAR(regno)
);

-- Create PARTICIPATED table with foreign keys
CREATE TABLE PARTICIPATED (
  driver_id INT,
  regno INT,
  report_number INT,
  damage_amount DECIMAL(10, 2),
  FOREIGN KEY (driver_id) REFERENCES PERSON(driver_id),
  FOREIGN KEY (regno) REFERENCES CAR(regno),
  FOREIGN KEY (report_number) REFERENCES ACCIDENT(report_number)
);

-- Insert data into PERSON table
INSERT INTO PERSON VALUES
/1 'John Doe' '123 Main St';
```

Run SQL

Available Tables

Output

SQL query successfully executed. However, the result set is empty.

ACCIDENT

report_number	accd_date	location
10	2023-01-05	Intersection A
11	2023-02-10	Highway B
12	2023-03-15	Street C
13	2023-04-20	Road D
14	2023-05-25	Avenue E

CAR

regno	model	year
101	Toyota Camry	2008
102	Honda Civic	2008
103	Ford Mustang	2009
104	Chevrolet Malibu	2010
105	Nissan Altima	2011

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders

Input



Run SQL

Available Tables

```
-- Insert data into PERSON table
INSERT INTO PERSON VALUES
(1, 'John Doe', '123 Main St'),
(2, 'Jane Smith', '456 Elm St'),
(3, 'Alice Johnson', '789 Oak St'),
(4, 'Bob Brown', '101 Pine St'),
(5, 'Emily Davis', '202 Maple St');
```

```
-- Insert data into CAR table
INSERT INTO CAR VALUES
(101, 'Toyota Camry', 2008),
(102, 'Honda Civic', 2008),
(103, 'Ford Mustang', 2009),
(104, 'Chevrolet Malibu', 2010),
(105, 'Nissan Altima', 2011);
```

```
-- Insert data into ACCIDENT table
INSERT INTO ACCIDENT VALUES
(10, '2023-01-05', 'Intersection A'),
(11, '2023-02-10', 'Highway B'),
(12, '2023-03-15', 'Street C'),
```

Output

SQL query successfully executed. However, the result set is empty.

ACCIDENT

report_number	accd_date	location
10	2023-01-05	Intersection A
11	2023-02-10	Highway B
12	2023-03-15	Street C
13	2023-04-20	Road D
14	2023-05-25	Avenue E

CAR

regno	model	year
101	Toyota Camry	2008
102	Honda Civic	2008
103	Ford Mustang	2009
104	Chevrolet Malibu	2010
105	Nissan Altima	2011

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Input



Run SQL

```
(10, '2023-01-05', 'Intersection A'),
(11, '2023-02-10', 'Highway B'),
(12, '2023-03-15', 'Street C'),
(13, '2023-04-20', 'Road D'),
(14, '2023-05-25', 'Avenue E');
```

```
-- Insert data into OWNS table
INSERT INTO OWNS VALUES
(1, 101),
(2, 102),
(3, 103),
(4, 104),
(5, 105);
```

```
-- Insert data into PARTICIPATED table
INSERT INTO PARTICIPATED VALUES
(1, 101, 12, 5000),
(2, 102, 12, 6000),
(3, 103, 13, 7000),
(4, 104, 14, 8000),
(5, 105, 14, 9000);
```

Output

SQL query successfully executed. However, the result set is empty.

Online SQL Editor

programiz.com/sql/online-compiler/

Guest

Programiz

Online SQL Editor

exness

Think Next Level Trading  
Think Exness

Upgrade now

Interactive SQL Course

ACCIDENT [-]

report\_number [int]

accid\_date [date]

location [varchar(100)]

CAR [-]

regno [int]

model [varchar(100)]

year [int]

Customers [-]

customer\_id [int]

first\_name [varchar(100)]

last\_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order\_id [integer]

item [varchar(100)]

amount [integer]

customer\_id [integer]

OWNS [-]

driver\_id [int]

regno [int]

PARTICIPATED [-]

driver\_id [int]

regno [int]

Input

```
(2, 102),
(3, 103),
(4, 104),
(5, 105);

-- Insert data into PARTICIPATED table
INSERT INTO PARTICIPATED VALUES
(1, 101, 12, 5000),
(2, 102, 12, 6000),
(3, 103, 13, 7000),
(4, 104, 14, 8000),
(5, 105, 14, 9000);

UPDATE PARTICIPATED
SET damage_amount = 25000
WHERE regno = [specific_regno] AND report_number = 12;
```

Output

Error: table PERSON already exists

Available Tables

5

105

PARTICIPATED

driver_id	regno	report_number	damage_amount
1	101	12	5000
2	102	12	6000
3	103	13	7000
4	104	14	8000
5	105	14	9000

PERSON

driver_id	name	address
1	John Doe	123 Main St
2	Jane Smith	456 Elm St
3	Alice Johnson	789 Oak St
4	Bob Brown	101 Pine St
5	Emily Davis	202 Maple St

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Online SQL Editor

programiz.com/sql/online-compiler/

Guest

Programiz

Online SQL Editor

exness

625,000+ traders choose  
Exness every month

Get started

Interactive SQL Course

ACCIDENT [-]

report\_number [int]

accid\_date [date]

location [varchar(100)]

CAR [-]

regno [int]

model [varchar(100)]

year [int]

Customers [-]

customer\_id [int]

first\_name [varchar(100)]

last\_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order\_id [integer]

item [varchar(100)]

amount [integer]

customer\_id [integer]

OWNS [-]

driver\_id [int]

regno [int]

PARTICIPATED [-]

driver\_id [int]

regno [int]

Input

```
(4, 104),
(5, 105);

-- Insert data into PARTICIPATED table
INSERT INTO PARTICIPATED VALUES
(1, 101, 12, 5000),
(2, 102, 12, 6000),
(3, 103, 13, 7000),
(4, 104, 14, 8000),
(5, 105, 14, 9000);

UPDATE PARTICIPATED
SET damage_amount = 25000
WHERE regno = [specific_regno] AND report_number = 12;
SELECT * FROM PARTICIPATED;
```

Output

Error: table PERSON already exists

Available Tables

5

105

PARTICIPATED

driver_id	regno	report_number	damage_amount
1	101	12	5000
2	102	12	6000
3	103	13	7000
4	104	14	8000
5	105	14	9000

PERSON

driver_id	name	address
1	John Doe	123 Main St
2	Jane Smith	456 Elm St
3	Alice Johnson	789 Oak St
4	Bob Brown	101 Pine St
5	Emily Davis	202 Maple St

Shippings

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Online SQL Editor

programiz.com/sql/online-compiler/

Guest

Programiz

Online SQL Editor

exness

625,000+ traders choose  
Exness every month

Get started

Interactive SQL Course

ACCIDENT [-]

report\_number [int]

accid\_date [date]

location [varchar(100)]

CAR [-]

regno [int]

model [varchar(100)]

year [int]

Customers [-]

customer\_id [int]

first\_name [varchar(100)]

last\_name [varchar(100)]

age [int]

country [varchar(100)]

Orders [-]

order\_id [integer]

item [varchar(100)]

amount [integer]

customer\_id [integer]

OWNS [-]

driver\_id [int]

regno [int]

PARTICIPATED [-]

driver\_id [int]

regno [int]

Input

```
INSERT INTO ACCIDENT (report_number, accid_date, location)
VALUES ([new_report_number], '[new_accid_date]', '[new_location]');
```

Output

Available Tables

5

105

PARTICIPATED

driver_id	regno	report_number	damage_amount
1	101	12	5000
2	102	12	6000
3	103	13	7000
4	104	14	8000
5	105	14	9000

PERSON

driver_id	name	address
1	John Doe	123 Main St
2	Jane Smith	456 Elm St

< Input

Run SQL

```
INSERT INTO ACCIDENT (report_number, accd_date, location)
VALUES ([new_report_number], '[new_accd_date]', '[new_location]');

SELECT COUNT(DISTINCT p.driver_id)
FROM PERSON p
INNER JOIN OWNS o ON p.driver_id = o.driver_id
INNER JOIN CAR c ON o.regno = c.regno
INNER JOIN PARTICIPATED pa ON o.regno = pa.regno
INNER JOIN ACCIDENT a ON pa.report_number = a.report_number
WHERE c.year = 2008;
```

< Input

Run SQL

```
SELECT COUNT(*)
FROM CAR c
INNER JOIN OWNS o ON c.regno = o.regno
INNER JOIN PARTICIPATED pa ON o.regno = pa.regno
INNER JOIN ACCIDENT a ON pa.report_number = a.report_number
WHERE c.model = '[specific_model]';
```

Output

COUNT(*)
0



main.py

```
1 import cx_Oracle
2
3 # Establish connection
4 connection = cx_Oracle.connect('username/password@hostname/service_name')
5
6 # Cursor
7 cursor = connection.cursor()
8
9 # Insert operation
10 def insert_employee(emp_id, emp_name, emp_salary):
11     try:
12         cursor.execute("INSERT INTO Employees (emp_id, emp_name, emp_salary) VALUES (:1, :2, :3)",
13                         (emp_id, emp_name, emp_salary))
14         connection.commit()
15         print("Employee inserted successfully.")
16     except cx_Oracle.Error as error:
17         print('Error occurred:', error)
18         connection.rollback()
19
20 # Update operation
21 def update_employee_salary(emp_id, new_salary):
22     try:
23         cursor.execute("UPDATE Employees SET emp_salary = :1 WHERE emp_id = :2", (new_salary, emp_id))
24         connection.commit()
25         print("Employee salary updated successfully.")
26     except cx_Oracle.Error as error:
27         print('Error occurred:', error)
28         connection.rollback()
29
30 # Trigger operation
31 # Assuming there is a trigger named 'payroll_trigger' that performs certain actions when a new
32 # employee is inserted
33 # You can execute a dummy insert statement to trigger the trigger
34 def trigger_dummy_insert():
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ModuleNotFoundError: No module named 'cx_Oracle'

=== Code Exited With Errors ===
```

poogleads.g.doubleclick.net/pcs/click?xai=AKAQjstoCrKhHv8RSLv1u7fgWbANZp1WPWihicGhVYHdhlmSeTPk...

BSE midcap