OneCompiler

EDITOR CHALLENGES ORGS MORE LOGIN

queries.sql + 42puen4bu

NEW MYSQL RUN

```sql
1
2  CREATE TABLE Categoria_de_Produtos (
3      id_categoria INT PRIMARY KEY AUTO_INCREMENT,
4      nome_categoria VARCHAR(100) NOT NULL
5
6      CREATE TABLE Produtos (
7      id_produto INT PRIMARY KEY AUTO_INCREMENT,
8      nome_produto VARCHAR(100) NOT NULL,
9      preco DECIMAL(10, 2) NOT NULL,
10     id_categoria INT,
11     FOREIGN KEY (id_categoria) REFERENCES Categoria_de_Produtos(id_categoria)
12     );
13
14     INSERÇÃO DE CATEGORIAS
15
16     INSERT INTO Categoria_de_Produtos (nome_categoria) VALUES ('Eletrônicos');
17 INSERT INTO Categoria_de_Produtos (nome_categoria) VALUES ('Móveis');
18 INSERT INTO Categoria_de_Produtos (nome_categoria) VALUES ('Roupas');
19 INSERT INTO Categoria_de_Produtos (nome_categoria) VALUES ('Alimentos');
20
21 INSERÇÃO DE PRODUTOS
22
23 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Televisão LED', 1500.00, 1)
24 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Sofá', 700.00, 2);
25 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Cadeira de Escritório', 25
26 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Camiseta', 30.00, 3);
27 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Calça Jeans', 60.00, 3);
28 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Maçã', 5.00, 4);
29 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Pão', 2.00, 4);
30 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Televisão Smart', 2000.00,
31 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Mesa de Jantar', 500.00, 2)
32 INSERT INTO Produtos (nome_produto, preco, id_categoria) VALUES ('Suéter', 80.00, 3);
33
34
35
```

STDIN

Input for the program (Optional)

Output:

```
ERROR 1064 (42000) at line 2: You have an error in your SQL
    id_produto INT PRIMARY KEY AUTO_INCREMENT,
    nome_' at line 5
```

MySQL online editor

Write, Run & Share MySQL queries online using OneCompiler's MySQL online editor and compiler for free. It's one of the robust, feature-rich online editor and compiler for MySQL. Getting started with the OneCompiler's MySQL editor is really simple and pretty fast. The editor shows sample boilerplate code when you choose language as 'MySQL' and start writing queries to learn and test online without worrying about tedious process of installation.

About MySQL

MySQL is a open-source, free and very popular relational database management system which is developed, distributed and supported by Oracle corporation.

Key Features:

- Open-source relational database management systems.
- Reliable, very fast and easy to use database server.
- Works on client-server model.
- Highly Secure and Scalable
- High Performance
- High productivity as it uses stored procedures, triggers, views to write a highly productive code.
- Supports large databases efficiently.
- Supports many operating systems like Linux*,CentOS*, Solaris*,Ubuntu*,Windows*, MacOS*,FreeBSD* and others.

Syntax help

Commands

1. CREATE

```
CREATE TABLE table_name (
                column1 datatype,
                column2 datatype,
                ....);
```

Example

```
CREATE TABLE EMPLOYEE (
  empId INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  dept TEXT NOT NULL
);
```

2. ALTER

```
ALTER TABLE Table_name ADD column_name datatype;
```

Example

```
INSERT INTO EMPLOYEE VALUES (0001, 'Dave', 'Sales');
```

3. TRUNCATE

```
TRUNCATE table table_name;
```

4. DROP

```
DROP TABLE table_name;
```

5. RENAME

```
RENAME TABLE table_name1 to new_table_name1;
```

6. COMMENT

Single-Line Comments:

```
 --Line1;
```

Multi-Line comments:

```
   /* Line1,
   Line2 */
```

DML Commands

1. INSERT

```
INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);
```

Note: Column names are optional.

Example

```
INSERT INTO EMPLOYEE VALUES (0001, 'Ava', 'Sales');
```

2. SELECT

```
SELECT column1, column2, ...
FROM table_name
[where condition];
```

Example

```
SELECT * FROM EMPLOYEE where dept ='sales';
```

3. UPDATE

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

Example

```
UPDATE EMPLOYEE SET dept = 'Sales' WHERE empId='0001';
```

4. DELETE

```
DELETE FROM table_name where condition;
```

Example

```
DELETE from EMPLOYEE where empId='0001';
```

Indexes

1. CREATE INDEX

```
  CREATE INDEX index_name on table_name(column_name);
```

- To Create Unique index:

```
  CREATE UNIQUE INDEX index_name on table_name(column_name);
```

2. DROP INDEX

```
DROP INDEX index_name ON table_name;
```

Views

1. Create a View

```
Creating a View:
CREATE VIEW View_name AS
Query;
```

2. How to call view

```
SELECT * FROM View_name;
```

3. Altering a View

```
ALTER View View_name AS
Query;
```

4. Deleting a View

```
DROP VIEW View_name;
```

Triggers

1. Create a Trigger

```
CREATE TRIGGER trigger_name trigger_time trigger_event
    ON tbl_name FOR EACH ROW [trigger_order] trigger_body
/* where
trigger_time: { BEFORE | AFTER }
trigger_event: { INSERT | UPDATE | DELETE }
trigger_order: { FOLLOWS | PRECEDES } */
```

2. Drop a Trigger

```
DROP TRIGGER [IF EXISTS] trigger_name;
```

Stored Procedures

1. Create a Stored Procedure

```
CREATE PROCEDURE sp_name(p1 datatype)
BEGIN
/*Stored procedure code*/
END;
```

2. How to call Stored procedure

```
CALL sp_name;
```

3. How to delete stored procedure

```
DROP PROCEDURE sp_name;
```

Joins

1. INNER JOIN

```
SELECT * FROM TABLE1 INNER JOIN TABLE2 where condition;
```

2. LEFT JOIN

```
SELECT * FROM TABLE1 LEFT JOIN TABLE2 ON condition;
```

3. RIGHT JOIN

```
SELECT * FROM TABLE1 RIGHT JOIN TABLE2 ON condition;
```

4. CROSS JOIN

```
SELECT select_list from TABLE1 CROSS JOIN TABLE2;
```