Agora vamos ver como criar tabelas no MySQL.

A instrução **CREATE TABLE** permite criar uma nova tabela em um banco de dados.

A seguir, ilustramos a sintaxe básica da instrução CREATE TABLE:

```
CREATE TABLE [IF NOT EXISTS] table_name(
    column_1_definition,
    column_2_definition,
    ...,
    table_constraints
) ENGINE=storage_engine;
```

Primeiro, você deve especificar o nome da tabela que deseja criar após as palavras-chave CREATE TABLE. O nome da tabela deve ser exclusivo dentro de um banco de dados. O IF NOT EXISTS é opcional. Ele permite que você verifique se a tabela que você cria já existe no banco de dados. Se for esse o caso, o MySQL ignorará toda a instrução e não criará nenhuma nova tabela.

Segundo, você deve especificar uma lista de colunas da tabela na seção column_list, as colunas são separadas por vírgulas.

Terceiro, você pode opcionalmente especificar o mecanismo de armazenamento para a tabela na cláusula ENGINE. Você pode usar qualquer mecanismo de armazenamento como InnoDB e MyISAM. Se você não declarar explicitamente um mecanismo de armazenamento, o MySQL usará o InnoDB por padrão.

O InnoDB se tornou o mecanismo de armazenamento padrão desde o MySQL versão 5.5. O mecanismo de armazenamento InnoDB traz muitos benefícios de um sistema de gerenciamento de banco de dados relacional, como transações ACID, integridade referencial e recuperação de falhas. Nas versões anteriores, o MySQL usava o MyISAM como o mecanismo de armazenamento padrão.

A seguir, é mostrada a sintaxe da definição de uma coluna:

```
column_name data_type(length) [NOT NULL] [DEFAULT value]
[AUTO INCREMENT] column constraint;
```

Aqui estão os detalhes:

- O nome da coluna especifica o nome da coluna. Cada coluna tem um tipo de dados específico e tamanho opcional, por exemplo, VARCHAR (255)
- A restrição NOT NULL garante que a coluna não contenha NULL. Além da restrição NOT NULL, uma coluna pode ter restrições adicionais, como CHECK e UNIQUE.
- O padrão especifica um valor padrão para a coluna.
- O AUTO_INCREMENT indica que o valor da coluna é incrementado por um automaticamente sempre que uma nova linha é inserida na tabela. Cada tabela possui no máximo uma coluna AUTO INCREMENT.

Após a lista de colunas, você pode definir restrições de tabela como <u>UNIQUE</u>, <u>CHECK</u>, <u>PRIMARY KEY</u> e <u>FOREIGN KEY</u>. Por exemplo, se você deseja definir uma coluna ou um grupo de colunas como a chave principal, use a seguinte sintaxe:

```
PRIMARY KEY (col1, col2,...)
```

Exemplos de instruções MySQL CREATE TABLE

Vamos ver alguns exemplos de criação de novas tabelas. A instrução a seguir cria uma nova tabela chamada tasks:

```
CREATE TABLE IF NOT EXISTS tasks (

task_id INT AUTO_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

start_date DATE,

due_date DATE,

status TINYINT NOT NULL,

priority TINYINT NOT NULL,

description TEXT,

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

ENGINE=INNODB;
```

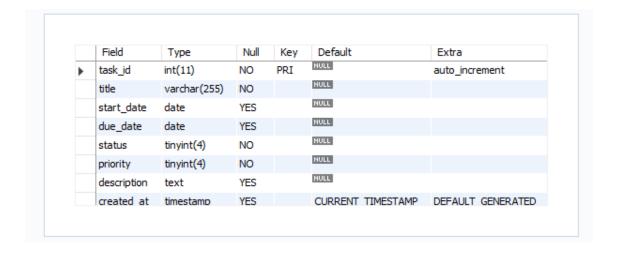
A tabela de tarefas possui as seguintes colunas:

- O tasks_id é uma coluna de incremento automático. Se você usar a instrução INSERT para inserir uma nova linha na tabela sem especificar um valor para a coluna tasks_id, o MySQL gerará automaticamente um número inteiro seqüencial para o tasks id a partir de 1.
- A coluna title é uma coluna de cadeia de caracteres variável cujo comprimento máximo é 255. Isso significa que você não pode inserir uma cadeia cujo comprimento seja maior que 255 nessa coluna. A restrição <u>NOT NULL</u> indica que a coluna não aceita NULL. Em outras palavras, você deve fornecer um valor não NULL ao inserir ou atualizar esta coluna.
- O start_date e due_date são colunas <u>DATE</u>. Como essas colunas não têm a restrição NOT NULL, elas podem armazenar NULL. A coluna start_date possui um valor padrão da data atual. Em outras palavras, se você não fornecer um valor para a coluna start_date ao inserir uma nova linha, a coluna start_date terá a data atual do servidor de banco de dados.
- O status e a priority são as colunas <u>TINYINT</u> que não permitem NULL. A coluna description é uma coluna de TEXT que aceita NULL.
- O created_at é uma coluna <u>TIMESTAMP</u> que aceita o horário atual como o valor padrão.

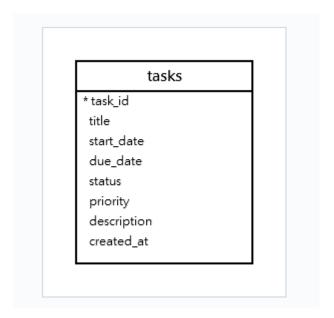
O tasks_id é a coluna da chave primária da coluna da tabela tasks. Isso significa que os valores na coluna tasks_id identificarão exclusivamente as linhas na tabela.

Depois de executar a instrução CREATE TABLE para criar a tabela taks, é possível visualizar sua estrutura usando a instrução DESCRIBE:

DESCRIBE tasks;



Esta imagem mostra o diagrama do banco de dados da tabela taks:



MySQL CREATE TABLE com um exemplo de chave primária de chave estrangeira. Suponha que cada tarefa tenha uma lista de verificação ou lista de tarefas. Para armazenar listas de verificação de tarefas, você pode criar uma nova tabela denominada checklists da seguinte maneira: CREATE TABLE IF NOT EXISTS checklists (

```
todo_id INT AUTO_INCREMENT,
task_id INT,
todo VARCHAR(255) NOT NULL,
is_completed BOOLEAN NOT NULL DEFAULT FALSE,
PRIMARY KEY (todo_id , task_id),
FOREIGN KEY (task_id)

REFERENCES tasks (task_id)
ON UPDATE RESTRICT ON DELETE CASCADE
```

A tabela checklists possui uma chave primária que consiste em duas colunas. Portanto, usamos uma restrição de tabela para definir a <u>primary key</u>:

```
PRIMARY KEY (todo_id , task_id)
```

Além disso, o task_id é a coluna de chave estrangeira que faz referência à coluna task_id da tabela tasks, usamos uma restrição de chave estrangeira para estabelecer esse relacionamento: FOREIGN KEY (task_id)

```
REFERENCES tasks (task_id)
ON UPDATE RESTRICT
ON DELETE CASCADE
```

Esta imagem ilustra a tabela checklists e seu relacionamento com a tabela de

