

Data Manipulation

In MySQL Database

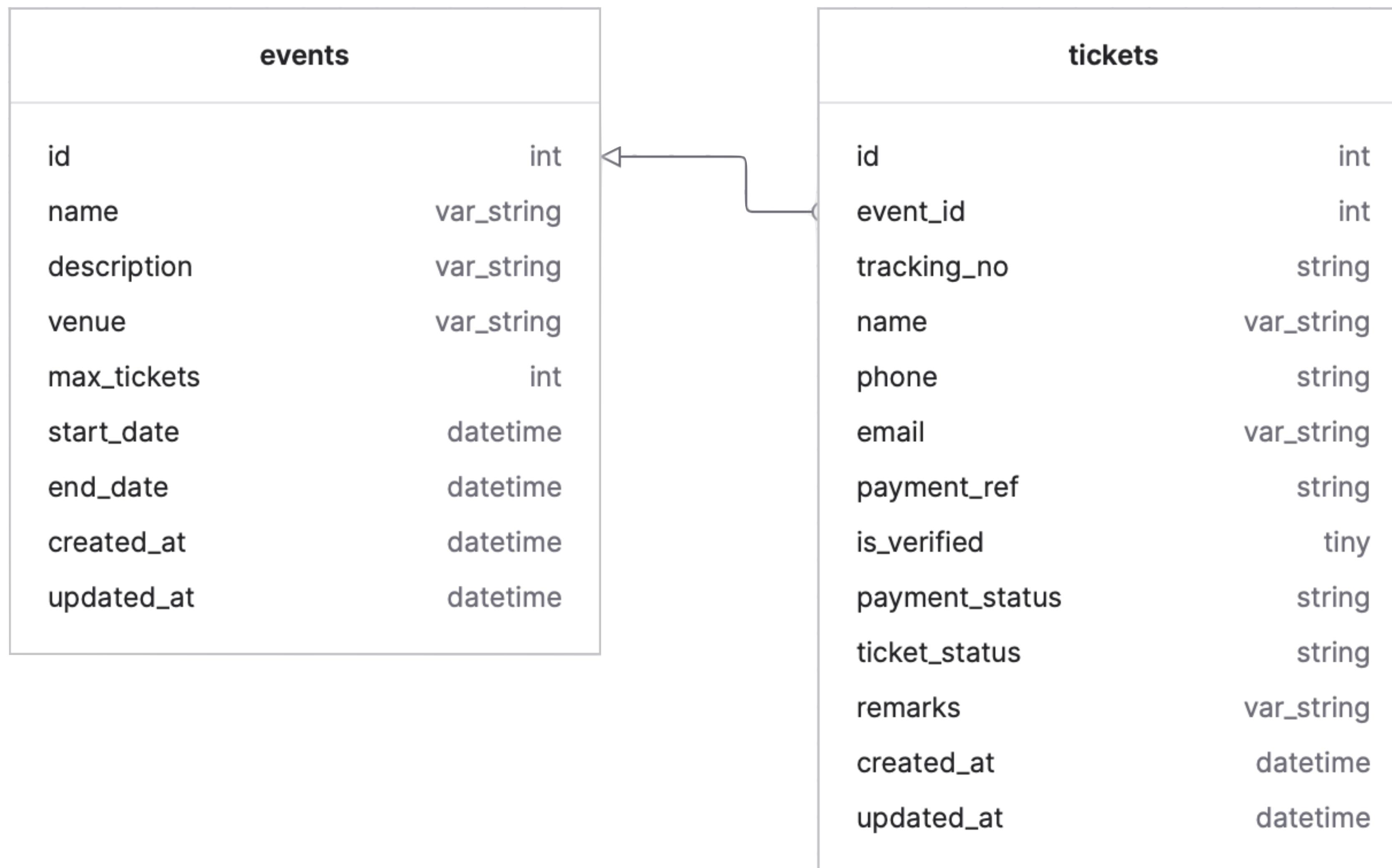
Example Scenario

Tech Event Registration

- Total Seats: 200
- We accept only MFS (bKash/Nogod/...)
- Verify payments manually by Admin
- We are in rush... some queries can be run from database directly (without UI).



Event Module Schema Diagram



Let's prepare the Tables

```
CREATE TABLE events (
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    `name` VARCHAR(50) NOT NULL,
    description VARCHAR(250),
    venue VARCHAR(50),
    max_tickets INT,
    start_date DATETIME,
    end_date DATETIME,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME ON UPDATE CURRENT_TIMESTAMP
);
```

```
CREATE TABLE tickets (
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    tracking_no CHAR(8) NOT NULL,
    `name` VARCHAR(50) NOT NULL,
    phone CHAR(15) NOT NULL,
    email VARCHAR(50),
    payment_ref CHAR(10),
    is_verified BOOLEAN DEFAULT FALSE,
    payment_status ENUM('pending', 'received', 'refunded', 'invalid') DEFAULT 'pending',
    ticket_status ENUM('booked', 'confirmed', 'cancelled') DEFAULT 'booked',
    remarks VARCHAR(250),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (event_id) REFERENCES events (id),
    UNIQUE INDEX unq_payment_ref (payment_ref),
    UNIQUE INDEX unq_tracking_no (tracking_no)
);
```



Create New Record

```
INSERT INTO {table_name} [({list, of, fields})]  
VALUES ({'list', 'of', 'values'});
```

```
INSERT INTO events [({list, of, fields})]  
VALUES ('list', 'of', 'values');
```

```
INSERT INTO events [({list, of, fields})]  
VALUES ('list', 'of', 'values');
```

events

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
#		NULL	NULL		NULL	NULL	CURRENT_TIMESTAMP

```
INSERT INTO events ( name, venue, max_tickets, start_date )
VALUES ('list', 'of', 'values');
```

events

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
#		NULL				NULL	CURRENT_TIMESTAMP

```
INSERT INTO events ( name, venue, max_tickets, start_date )  
VALUES ('list', 'of', 'values');
```

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
#		NULL				NULL	CURRENT_TIMESTAMP

```
INSERT INTO events ( name, venue, max_tickets, start_date )
VALUES ('Tech Talk 2023', 'Online', 200, '2023-12-05 15:00:00');
```

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
#		NULL				NULL	CURRENT_TIMESTAMP

```
INSERT INTO events ( name, venue, max_tickets, start_date )
VALUES ('Tech Talk 2023', 'Online', 200, '2023-12-05 15:00:00');
```

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
#		NULL				NULL	CURRENT_TIMESTAMP

```
INSERT INTO events ( name, venue, max_tickets, start_date )
VALUES ('Tech Talk 2023', 'Online', 200, '2023-12-05 15:00:00');
```

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
2	Tech Talk 2023	<i>NULL</i>	Online	200	2023-12-05 15:00:00	<i>NULL</i>	2023-11-28 09:20:49

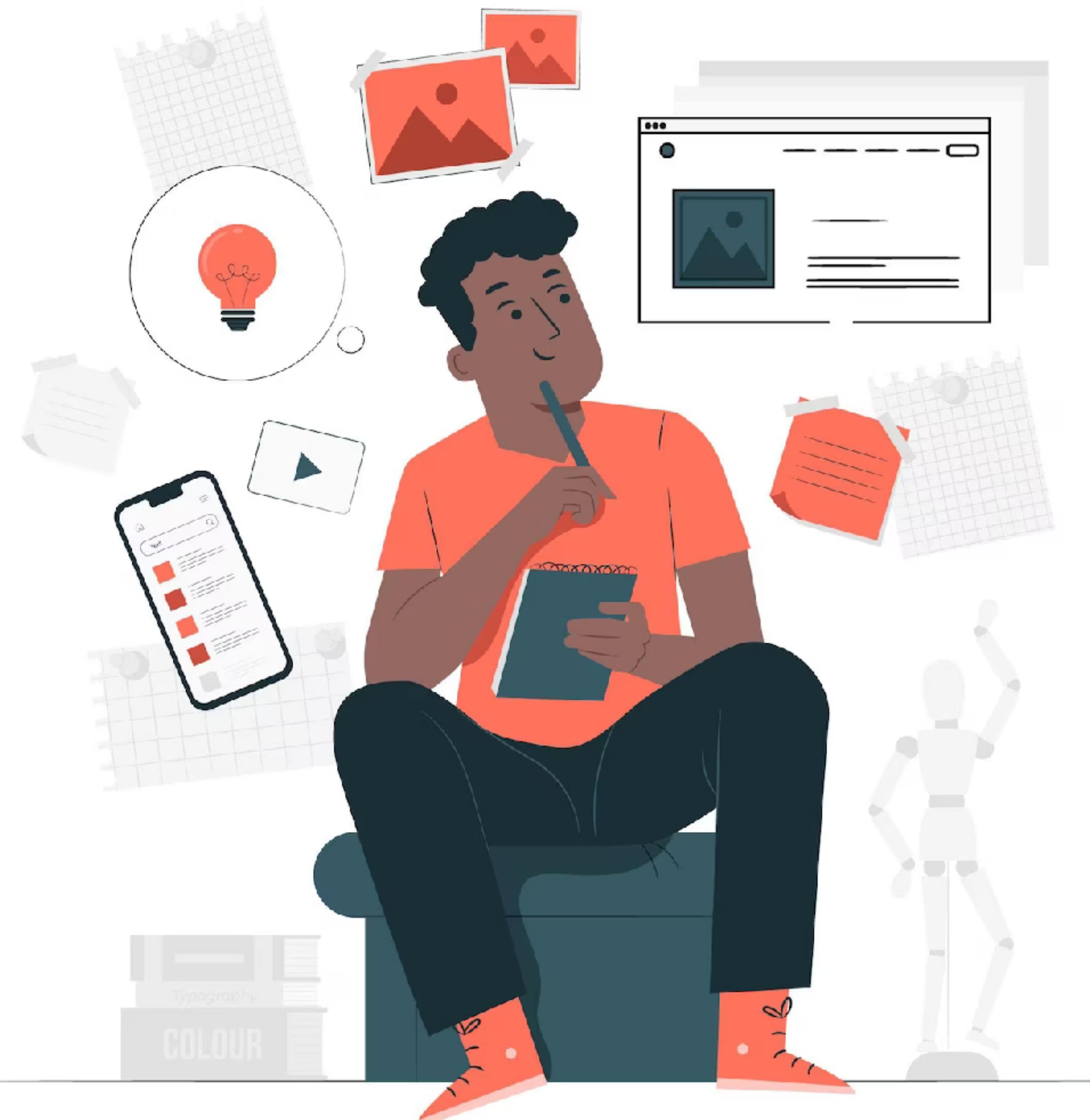
```
INSERT INTO events ( name, venue, max_tickets, start_date )
VALUES ('Tech Talk 2023', 'Online', 200, '2023-12-05 15:00:00');
```

events

id	name	description	venue	max_tickets	start_date	end_date	created_at
1	Test Event	Some Desc....	ABC Convention Hall	100	2023-11-20 07:05:49	2023-11-20 07:05:49	2023-11-20 07:05:49
2	Tech Talk 2023	NULL	Online	200	2023-12-05 15:00:00	NULL	2023-11-28 09:20:49

Inserting Multiple Records

```
INSERT INTO {table_name} [({list, of, fields})]  
VALUES ({'list', 'of', 'values'}),  
({'list', 'of', 'values'}),  
({'list', 'of', 'values'});
```



What can go wrong?

Ensure Input For ALL* fields

Which fields can be ignored?

- A. Configured for Generated Values (e.g. `AUTO_INCREMENT`)
- B. `NULLable` fields
- C. Have a `DEFAULT` value

Ensure Input For ALL* fields

```
CREATE TABLE events (
```

```
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
```

```
    `name` VARCHAR(50) NOT NULL,
```

```
    description VARCHAR(250),
```

```
    venue VARCHAR(50),
```

```
    max_tickets INT,
```

```
    start_date DATETIME,
```

```
    end_date DATETIME,
```

```
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
```

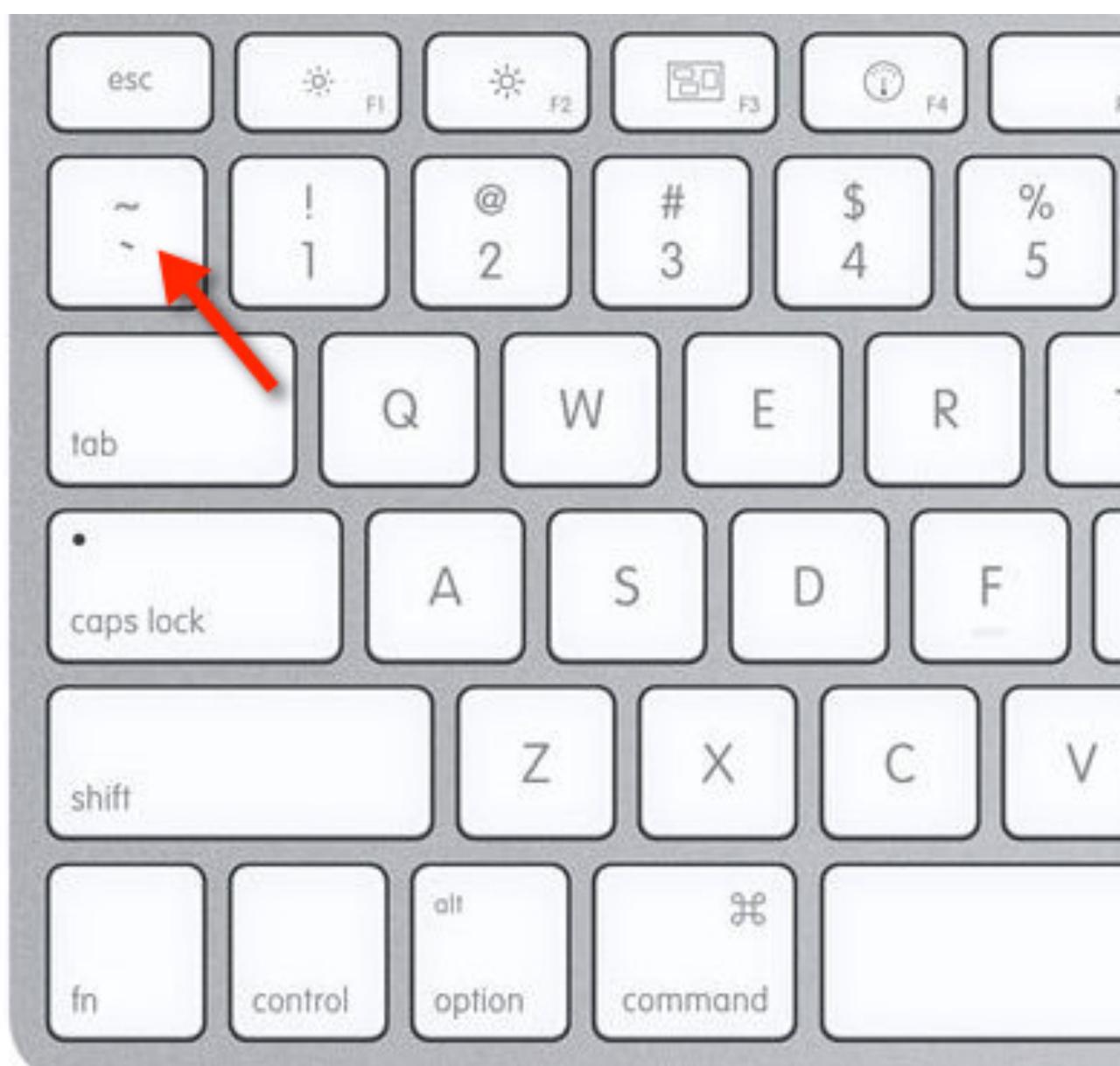
```
    updated_at DATETIME ON UPDATE CURRENT_TIMESTAMP
```

```
);
```

Can it be Ignored?	Why?
✓	AUTO_INCREMENT
✗	
✓	NULLABLE
✓	DEFAULT
✓	NULLABLE

Quoting NAMEs

- ☑ Backticks (`) are used around database name, table name, column name
- ☑ They are required if
 - ▶ Your database/table/column names contain whitespace characters
 - ▶ The name matches a SQL keyword or reserved word



Quoting NAMEs

CREATE DATABASE store room;	✗ Space in the name
CREATE DATABASE `store room`;	✓
CREATE DATABASE store_room;	✓
CREATE TABLE student profile (id INT);	✗ Space in the name
CREATE TABLE `student profile` (id INT);	✓
CREATE TABLE student_profile (id INT);	✓
CREATE TABLE ANALYZE (id INT);	Reserved word, will not work
INSERT INTO `ANALYZE` (`CHARACTER`) VALUES('Ruhul Amin');	✓

Some other things to check

If an Update/Insert fails

- Correct Syntax
 - ▶ Check for comma after each fields and indexes
 - ▶ No comma after the last one
 - ▶ All brackets are closed properly
- Valid Table and Column names (quoted if necessary)
- Sufficient Privileges to create record
- Matching Data Types, lengths, and ranges
- Constraints and Triggers (e.g., Duplicate value for UNIQUE column)
- Sufficient Storage Space

Handling Special Characters



String Literals

Quoting Strings

Strings and Dates need to be quoted

- Use single quote (') to quote strings and dates.
- Double quote (") can be used sometimes*.
- Multiple string side by side will be concatenated to a single string.
- MySQL `QUOTE()` function can be used to quote string.

* If `ANSI_QUOTES` mode is not enabled

String Literals

Quotation in Strings

- In a string quoted with single quote ('):
 - Double quote don't require any special treatment
 - Single quote can be placed as two single quote ("")
 - Single quote can be escaped with backslash (\')

String Literals

Quotation in Strings

- In a string quoted with single quote ('):

- Double quote don't require any special treatment
- Single quote can be placed as two single quote ("")
- Single quote can be escaped with backslash (\')

```
SELECT 'hello', '"hello"', """hello""", 'hel''lo', '\hello';
```

hello	"hello"	"""hello"""	hel'lo	'hello
-------	---------	-------------	--------	--------

String Literals

Quotation in Strings

- In a string quoted with double quote ("):
 - Single quote don't require any special treatment
 - Double quote can be placed as two double quoted ("")
 - Double quote can be escaped with backslash (\")

String Literals

Quotation in Strings

- In a string quoted with double quote ("):

- Single quote don't require any special treatment
- Double quote can be placed as two double quoted ("")
- Double quote can be escaped with backslash (\")

```
SELECT "hello", "'hello'", '''hello''', "hel""lo", "\hello";
```

hello	'hello'	"hello"	hel"lo	"hello
-------	---------	---------	--------	--------

String Literals

Escaping Other Special Characters

Escape Character Sequence	Character Represented by Sequence
\'	A single quote (') character
\"	A double quote (") character
\b	A backspace character
\n	A newline (linefeed) character
\r	A carriage return character
\t	A tab character
\\\	A backslash (\) character
\%	A % character (in search pattern)
_	A _ character (in search pattern)

Let's start Selling Tickets

Secure your seat!

Name

Phone

Email

Transaction ID

[What is this?](#)

Let's start Selling Tickets

Secure your seat!

Name

Phone

Email

Transaction ID
[What is this?](#)

```
INSERT INTO tickets (event_id, `name`, phone, email, payment_ref, tracking_no)
VALUES (1, 'Abdul Karim', '01712345678', 'karim@example.com', 'AXBY4D23',
        UPPER(LEFT(UUID(), 6)));
```

Let's start Selling Tickets

How application does it?

```
$conn→query("INSERT INTO tickets (event_id, `name`, phone, email, payment_ref,  
tracking_no)  
VALUES ('{$_POST['event_id']}',  
'{$_POST['name']}',  
'{$_POST['phone']}',  
'{$_POST['email']}',  
'{$_POST['payment_ref']}',  
UPPER(LEFT(UUID(), 6)))");
```

Let's start Selling Tickets

How application does it?

```
$conn→query("INSERT INTO tickets (event_id, `name`, phone, email, payment_ref,  
tracking_no)  
VALUES ('{$_POST['event_id']}',  
'{$_POST['name']}',  
'{$_POST['phone']}',  
'{$_POST['email']}',  
'{$_POST['payment_ref']}',  
UPPER(LEFT(UUID(), 6)))");
```

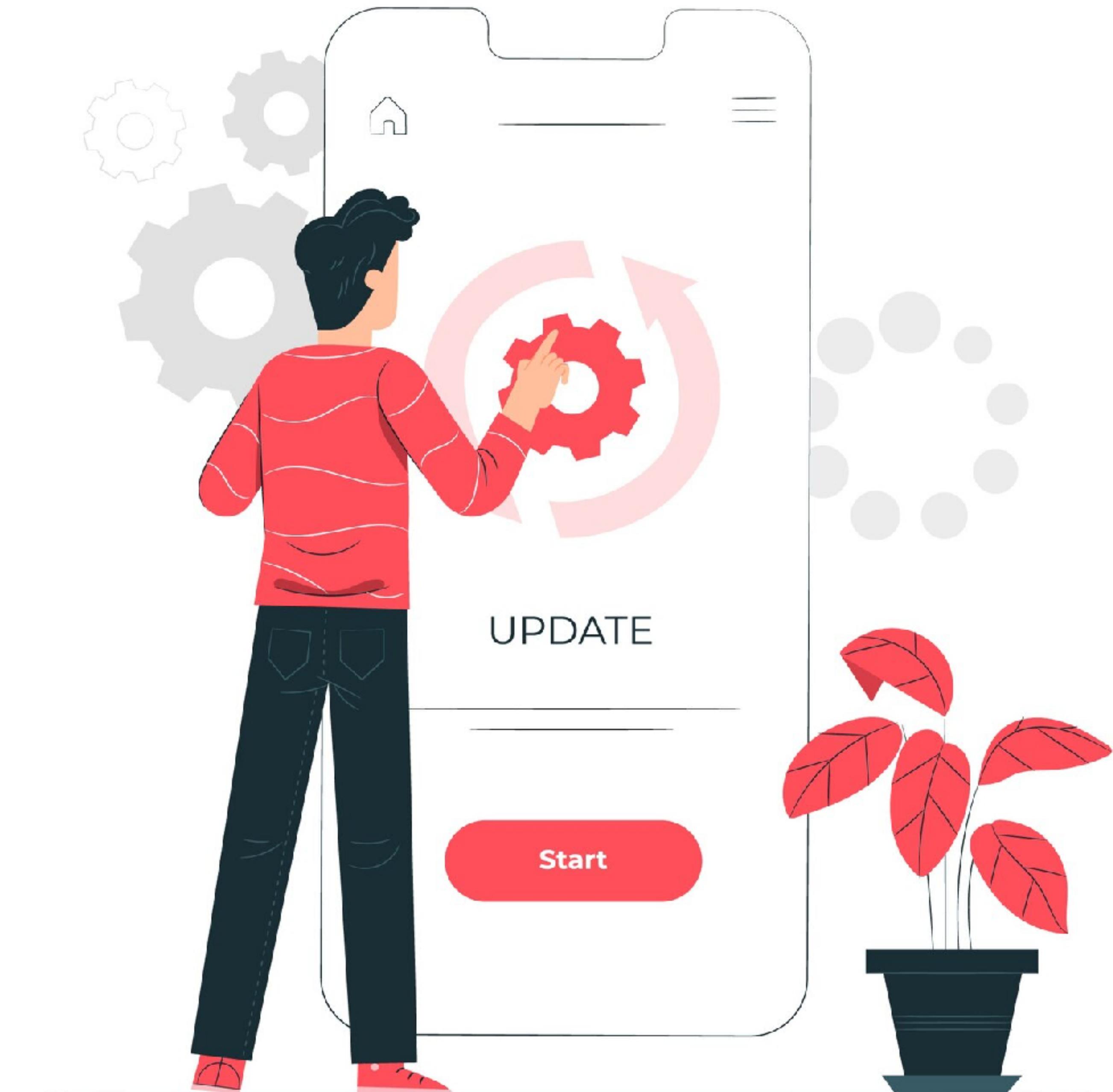
✗ Beware of SQL Injection

Let's start Selling Tickets

- Take precaution to prevent SQL Injection
- Handle user inputs with security in mind

```
$stmt = $db->prepare('INSERT INTO tickets (`name`, phone, email, payment_ref, tracking_no)  
VALUES (:name, :phone, :email, :trxid, :tracking)');  
  
$stmt->execute([  
    ':name' => $name,  
    ':phone' => $phone,  
    ':email' => $email,  
    ':trxid' => $trx_id,  
    ':tracking' => generateTicketNumber()  
]);
```

Updating Records



Verify Payments

Verify Payment

Name Abdul Karim

Phone 01712345678

Email karim@example.com

Transaction ID AXBY4D23

Verified Invalid

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

Verify Payment

Name Abdul Karim

Phone 01712345678

Email karim@example.com

Transaction ID AXBY4D23

Verified Invalid

is_verified	payment_status	ticket_status
1	received	confirmed

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

Verify Payment

Name Abdul Karim
Phone 01712345678
Email karim@example.com
Transaction ID AXBY4D23

Verified **Invalid**

is_verified	payment_status	ticket_status
1	received	confirmed
0	invalid	cancelled

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

```
UPDATE {table_name}  
SET {column1 = value1}[, {column2 = value2}...]  
WHERE {condition};
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

UPDATE tickets

SET {column1 = value1}[, {column2 = value2}...]

WHERE {condition};

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

```
UPDATE tickets
SET is_verified = true,
    payment_status = 'received',
    ticket_status = 'confirmed'
WHERE {condition};
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Verify Payments

```
UPDATE tickets
SET is_verified = true,
    payment_status = 'received',
    ticket_status = 'confirmed'
WHERE id = 1;
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	0	pending	booked
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

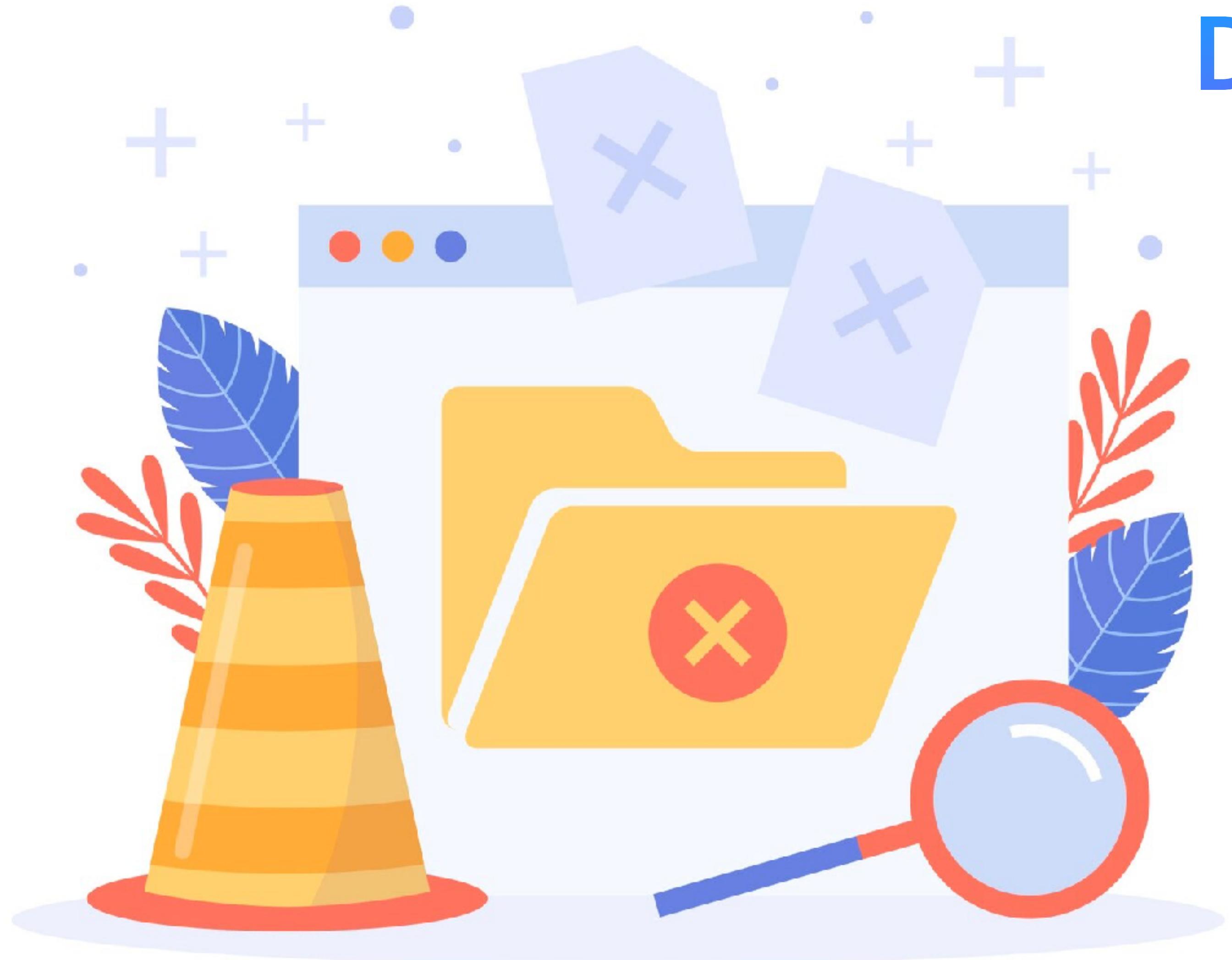
Verify Payments

```
UPDATE tickets
SET is_verified = true,
    payment_status = 'received',
    ticket_status = 'confirmed'
WHERE id = 1;
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	1	received	confirmed
2	1	C4B106	Abdur Rahim	01712345111	rahim@example.com	WEXBY42E	0	pending	booked

Deleting Records



Ticket Moderation

Ticket Management

Name	Dusto Chele	Status	Booked
Phone	00000	Payment	Pending
Email	aa@b.com	Remarks	<input type="text"/>
Transaction ID	I don't kn		

UpdateDeleteCancel

Deleting Record

```
DELETE FROM {table_name}  
WHERE {condition};
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	1	received	confirmed
4	1	C4B106	Dusto Chele	0000	aa@b.com	I don't kn	0	pending	booked

Deleting Record

```
DELETE FROM tickets  
WHERE {condition};
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	1	received	confirmed
4	1	C4B106	Dusto Chele	0000	aa@b.com	I don't kn	0	pending	booked

Deleting Record

```
DELETE FROM tickets  
WHERE id = 4 ;
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	1	received	confirmed
4	1	C4B106	Dusto Chele	0000	aa@b.com	I don't kn	0	pending	booked

Deleting Record

```
DELETE FROM tickets  
WHERE id = 4 ;
```

tickets

id	event_id	tracking_no	name	phone	email	payment_ref	is_verified	payment_status	ticket_status
1	1	30C7FD	Abdul Karim	01712345678	karim@example.com	AXBY4D23	1	received	confirmed

Handling Data-Type and Length Mismatch



Exceeding String Length

Ticket Management

Name	Dusto Chele	Status	Booked ▾
Phone	00000	Payment	Pending ▾
Email	aa@b.com	Remarks	<input type="text"/>
Transaction ID	I don't know		

Truncated Text!

[Update](#) [Delete](#) [Cancel](#)

```
CREATE TABLE data_test (id TINYINT NOT NULL PRIMARY KEY, title CHAR(5));  
INSERT INTO data_test VALUES(1, 'ANIS');  
INSERT INTO data_test VALUES(1, 'ANIS UDDIN AHMAD');  
INSERT INTO data_test VALUES(500, 'ANIS');
```

```
CREATE TABLE data_test (id TINYINT NOT NULL PRIMARY KEY, title CHAR(5));  
  
INSERT INTO data_test VALUES(1, 'ANIS');  
  
INSERT INTO data_test VALUES(1, 'ANIS UDDIN AHMAD');  
  
INSERT INTO data_test VALUES(500, 'ANIS');
```

	Strict Mode Enabled	Strict Mode Disabled
Exceeded String Length	Error	Warning, Truncate
Exceeded Numeric Range	Error	Warning, Try to fit

Managing MySQL STRICT mode

How to find?

```
SELECT @@GLOBAL.sql_mode;
```

```
SELECT @@SESSION.sql_mode;
```

```
SHOW VARIABLES LIKE '%sql_mode%';
```

How to Set?

```
SET sql_mode = ''; – disable all modes
```

```
SET sql_mode = @@GLOBAL.sql_mode;
```

```
SET sql_mode = 'STRICT_ALL_TABLES';
```

Vital modes to check for: **STRICT_ALL_TABLES, STRICT_TRANS_TABLES***

* Allows NULL sometimes, Transactional operations only

Errors and Warnings

Finding issues in the last query

```
SHOW WARNINGS;
```

```
SELECT @@warning_count;
```

```
SHOW ERRORS;
```

```
SELECT @@error_count;
```

Questions?

