

## Assignment:- 04

### Data types

#### Instructions:

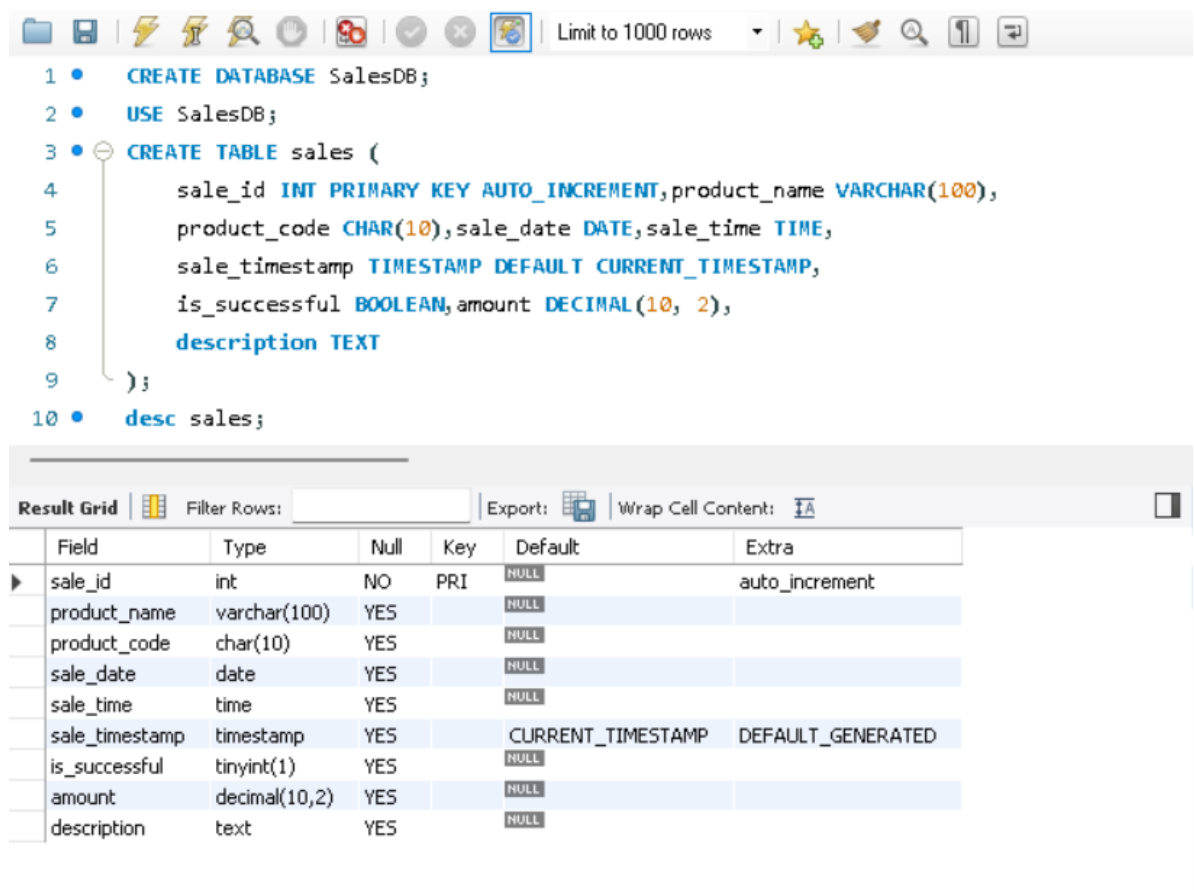
Please share your answers filled in line in the Word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name:** ULLI VENKATA SAI KUMAR **Batch ID:** 04072024HYD10AM

**Topic:** Introduction to Database

1. Create a database with a sales table containing data types like int, varchar, char ,date, time, timestamp, Boolean, decimal, text ?



```

1 • CREATE DATABASE SalesDB;
2 • USE SalesDB;
3 • CREATE TABLE sales (
4     sale_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHAR(100),
5     product_code CHAR(10), sale_date DATE, sale_time TIME,
6     sale_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
7     is_successful BOOLEAN, amount DECIMAL(10, 2),
8     description TEXT
9 );
10 • desc sales;
  
```

Field	Type	Null	Key	Default	Extra
sale_id	int	NO	PRI	<b>NULL</b>	auto_increment
product_name	varchar(100)	YES		<b>NULL</b>	
product_code	char(10)	YES		<b>NULL</b>	
sale_date	date	YES		<b>NULL</b>	
sale_time	time	YES		<b>NULL</b>	
sale_timestamp	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
is_successful	tinyint(1)	YES		<b>NULL</b>	
amount	decimal(10,2)	YES		<b>NULL</b>	
description	text	YES		<b>NULL</b>	

## Assignment:- 04

### Data types

2. Insert 10 random values in the table ?

SQL File 7\* SQL File 8\* SQL File 10\* SQL File 11\* SQL File 12\* SQL File 9\* SQL File 9\* x

Limit to 1000 rows

```

12 • INSERT INTO sales (product_name, product_code, sale_date, sale_time, is_successful, amount, description)
13 VALUES
14 ('Smartphone', 'PROD001', '2024-08-10', '14:30:00', TRUE, 499.99, 'High-end smartphone with 128GB storage'),
15 ('Laptop', 'PROD002', '2024-08-11', '09:15:00', FALSE, 899.50, '15-inch laptop with 16GB RAM'),
16 ('Headphones', 'PROD003', '2024-08-12', '16:45:00', TRUE, 79.99, 'Noise-cancelling over-ear headphones'),
17 ('Smartwatch', 'PROD004', '2024-08-13', '11:00:00', TRUE, 199.99, 'Waterproof smartwatch with GPS'),
18 ('Tablet', 'PROD005', '2024-08-14', '13:20:00', FALSE, 299.99, '10-inch tablet with detachable keyboard'),
19 ('Camera', 'PROD006', '2024-08-15', '10:30:00', TRUE, 650.00, 'DSLR camera with 24MP sensor'),
20 ('Gaming Console', 'PROD007', '2024-08-16', '17:00:00', TRUE, 499.00, 'Next-gen gaming console with VR support'),
21 ('Router', 'PROD008', '2024-08-17', '12:45:00', FALSE, 120.49, 'High-speed wireless router with dual-band support'),
22 ('Monitor', 'PROD009', '2024-08-18', '15:10:00', TRUE, 189.99, '27-inch 4K monitor with HDR'),
23 ('Keyboard', 'PROD010', '2024-08-19', '14:55:00', TRUE, 49.99, 'Mechanical keyboard with RGB lighting');
24 • select * from sales;
  
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

sale_id	product_name	product_code	sale_date	sale_time	sale_timestamp	is_successful	amount	description
1	Smartphone	PROD001	2024-08-10	14:30:00	2024-08-17 07:21:19	1	499.99	High-end smartphone with 128GB storage
2	Laptop	PROD002	2024-08-11	09:15:00	2024-08-17 07:21:19	0	899.50	15-inch laptop with 16GB RAM
3	Headphones	PROD003	2024-08-12	16:45:00	2024-08-17 07:21:19	1	79.99	Noise-cancelling over-ear headphones
4	Smartwatch	PROD004	2024-08-13	11:00:00	2024-08-17 07:21:19	1	199.99	Waterproof smartwatch with GPS
5	Tablet	PROD005	2024-08-14	13:20:00	2024-08-17 07:21:19	0	299.99	10-inch tablet with detachable keyboard
6	Camera	PROD006	2024-08-15	10:30:00	2024-08-17 07:21:19	1	650.00	DSLR camera with 24MP sensor
7	Gaming Console	PROD007	2024-08-16	17:00:00	2024-08-17 07:21:19	1	499.00	Next-gen gaming console with VR support
8	Router	PROD008	2024-08-17	12:45:00	2024-08-17 07:21:19	0	120.49	High-speed wireless router with dual-band support
9	Monitor	PROD009	2024-08-18	15:10:00	2024-08-17 07:21:19	1	189.99	27-inch 4K monitor with HDR
10	Keyboard	PROD010	2024-08-19	14:55:00	2024-08-17 07:21:19	1	49.99	Mechanical keyboard with RGB lighting

3. Change the data type of the existing column from DECIMAL (10,2) to FLOAT ?

25 • **ALTER TABLE sales MODIFY COLUMN amount FLOAT;**

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Field	Type	Null	Key	Default	Extra
sale_id	int	NO	PRI	NULL	auto_increment
product_name	varchar(100)	YES		NULL	
product_code	char(10)	YES		NULL	
sale_date	date	YES		NULL	
sale_time	time	YES		NULL	
sale_timestamp	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
is_successful	tinyint(1)	YES		NULL	
amount	float	YES		NULL	
description	text	YES		NULL	

## Assignment:- 04

### Data types

4. Change the data type of the existing column from Text to Varchar?

```
26 • ALTER TABLE sales MODIFY COLUMN description varchar(255);
27
28
29
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
sale_id	int	NO	PRI	<b>NULL</b>	auto_increment	
product_name	varchar(100)	YES		<b>NULL</b>		
product_code	char(10)	YES		<b>NULL</b>		
sale_date	date	YES		<b>NULL</b>		
sale_time	time	YES		<b>NULL</b>		
sale_timestamp	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED	
is_successful	tinyint(1)	YES		<b>NULL</b>		
amount	float	YES		<b>NULL</b>		
description	varchar(255)	YES		<b>NULL</b>		

5. What is “BLOB” Data Type in SQL, what are different types of BLOB?

**BLOB** stands for **Binary Large Object**. It is a data type in SQL used to store large amounts of binary data, such as images, audio, video, or any other type of multimedia files, as well as large documents. Unlike text data types (VARCHAR, TEXT), which store character data, BLOBs are specifically designed to handle binary data, which can be much larger and more complex.

#### **Different Types of BLOBs:**

The specific types of BLOBs vary depending on the SQL database system (e.g., MySQL, PostgreSQL). Here’s an overview of the most common BLOB types:

#### **1. TINYBLOB**

- **Size:** Up to 255 bytes.
- **Usage:** Suitable for storing very small binary objects, such as small icons or thumbnails.

#### **2. BLOB**

- **Size:** Up to 65,535 bytes (64 KB).

## **Assignment:- 04**

### **Data types**

- **Usage:** Typically used for medium-sized binary objects, like small images or short audio clips.

#### **3. MEDIUMBLOB**

- **Size:** Up to 16,777,215 bytes (16 MB).
- **Usage:** Used for larger binary objects, such as medium-sized images, longer audio files, or short video clips.

#### **4. LONGBLOB**

- **Size:** Up to 4,294,967,295 bytes (4 GB).
- **Usage:** Intended for very large binary objects, such as full-length movies, high-resolution images, or extensive binary data.

#### **6. Write different character data types and different numerical data types?**

##### **Character Data Types:-**

Character data types are used to store text or string data. Different SQL databases (e.g., MySQL, PostgreSQL, SQL Server) may have slight variations, but the following are the most commonly used character data types:

##### **1. CHAR (Fixed-Length)**

- **Description:** Stores fixed-length strings. If the input data is shorter than the specified length, the remaining space is padded with spaces.
- **Example:** CHAR(10) will always store 10 characters.
- **Use Case:** Useful when the data size is consistent, like storing country codes (e.g., "USA", "CAN").

##### **2. VARCHAR (Variable-Length)**

- **Description:** Stores variable-length strings. Unlike CHAR, it only uses as much space as needed to store the actual data.
- **Example:** VARCHAR(100) can store up to 100 characters but uses only the space required for the entered string.
- **Use Case:** Commonly used for names, emails, and any text data where length varies.

## **Assignment:- 04**

### **Data types**

#### **3. TEXT (Large Text)**

- **Description:** Stores large amounts of text data. The maximum length varies by database but typically allows for much more data than VARCHAR.
- **Example:** TEXT can store long descriptions, articles, or any substantial body of text.
- **Use Case:** Ideal for storing large text fields, such as comments or articles.

#### **4. TINYTEXT, MEDIUMTEXT, LONGTEXT (Variants in MySQL)**

- **TINYTEXT:** Up to 255 bytes.
- **MEDIUMTEXT:** Up to 16 MB.
- **LONGTEXT:** Up to 4 GB.
- **Use Case:** Used when you need to store varying sizes of text data, from very small to very large.

#### **Numerical Data Types:-**

Numerical data types are used to store numbers. These types can be divided into integer and floating-point types, as well as exact numeric types like DECIMAL.

##### **Integer Data Types**

#### **1. TINYINT**

- **Description:** Stores very small integers.
- **Range:** -128 to 127 (signed) or 0 to 255 (unsigned).
- **Use Case:** Useful for storing small numbers, like boolean flags or small counters.

#### **2. SMALLINT**

- **Description:** Stores small integers.
- **Range:** -32,768 to 32,767 (signed) or 0 to 65,535 (unsigned).
- **Use Case:** Suitable for larger counters, like IDs with a small range.

#### **3. MEDIUMINT (MySQL Specific)**

- **Description:** Stores medium-sized integers.
- **Range:** -8,388,608 to 8,388,607 (signed) or 0 to 16,777,215 (unsigned).

## **Assignment:- 04**

### **Data types**

- **Use Case:** Useful for storing moderate-sized numeric values.

#### **4. INT or INTEGER**

- **Description:** Stores standard integers.
- **Range:** -2,147,483,648 to 2,147,483,647 (signed) or 0 to 4,294,967,295 (unsigned).
- **Use Case:** Most common type for IDs, counters, and general-purpose integers.

#### **5. BIGINT**

- **Description:** Stores large integers.
- **Range:** -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (signed).
- **Use Case:** Ideal for very large numbers, like high-value counters or large IDs.

### **Floating-Point Data Types:-**

#### **1. FLOAT**

- **Description:** Stores single-precision floating-point numbers.
- **Use Case:** Used for storing approximate values with floating-point precision.

#### **2. DOUBLE**

- **Description:** Stores double-precision floating-point numbers.
- **Use Case:** Used when higher precision is required for floating-point arithmetic.

#### **3. REAL (Alias for FLOAT or DOUBLE depending on the SQL dialect)**

- **Description:** Implementation varies; often used as an alias for DOUBLE in some databases.
- **Use Case:** Same as FLOAT or DOUBLE.

### **Exact Numeric Types:-**

## **Assignment:- 04**

### **Data types**

#### **1. DECIMAL or NUMERIC**

- **Description:** Stores exact numeric values with a fixed number of decimal places.
- **Syntax:** DECIMAL(precision, scale), where precision is the total number of digits, and scale is the number of digits after the decimal point.
- **Example:** DECIMAL(10, 2) stores numbers up to 10 digits long, with 2 digits after the decimal point.
- **Use Case:** Ideal for financial data where exact precision is necessary, like currency.

#### **Summary**

- **Character Data Types:** CHAR, VARCHAR, TEXT (with variants like TINYTEXT, MEDIUMTEXT, LONGTEXT).
- **Numerical Data Types:** TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, FLOAT, DOUBLE, DECIMAL (or NUMERIC).