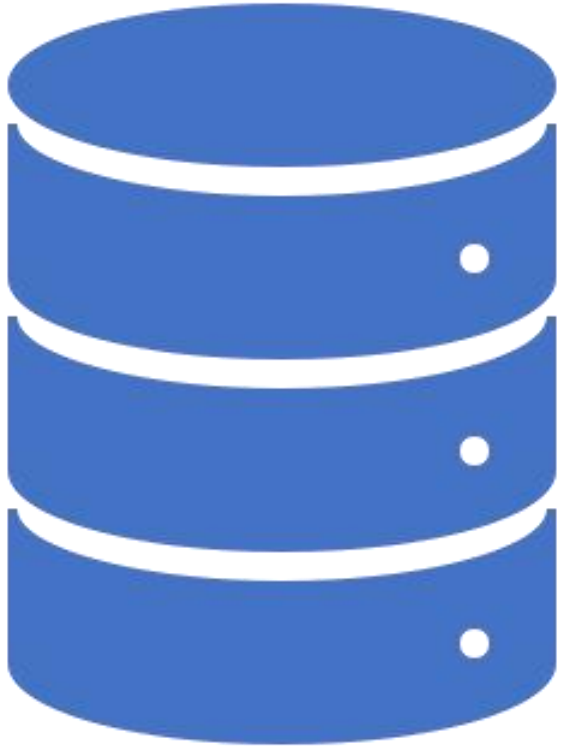
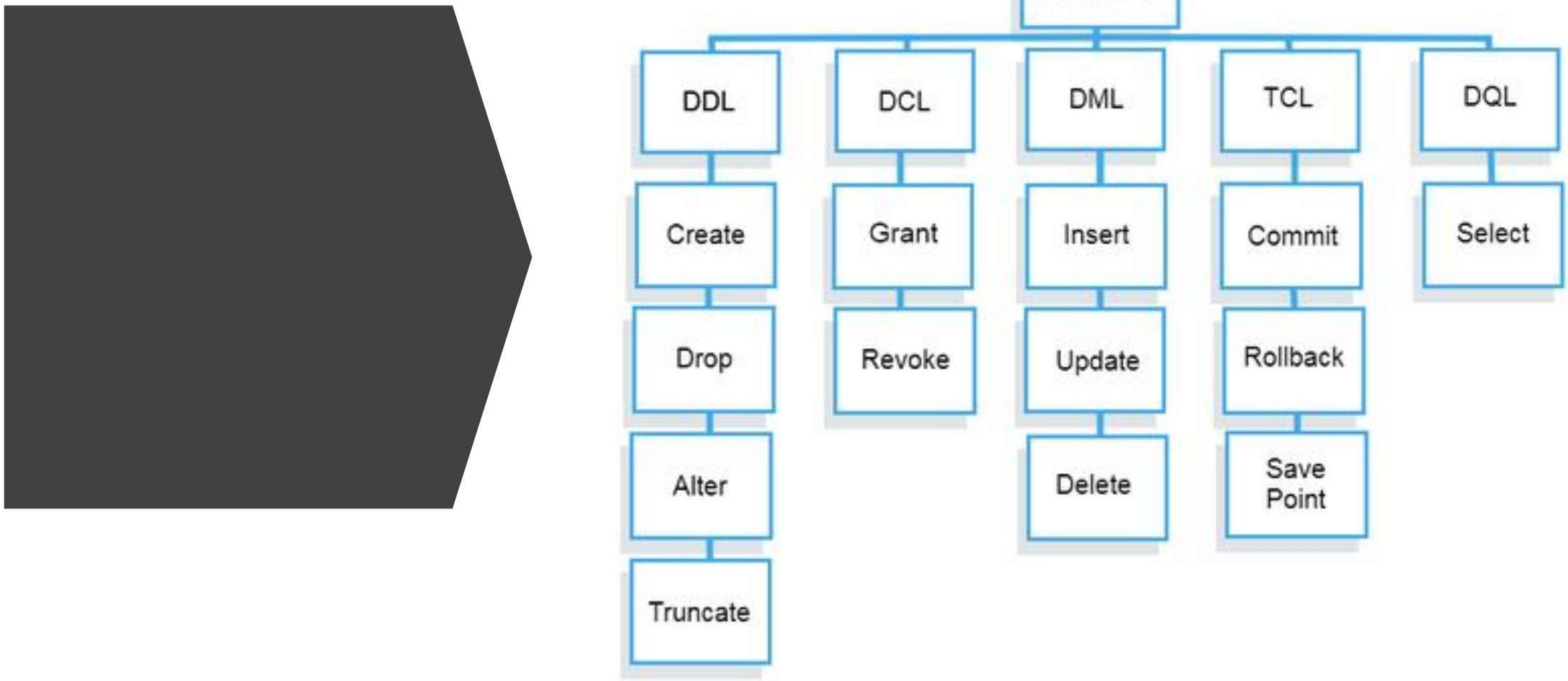
SQL Commands



- SQL Commands can be majorly classified in five categories.
  - 1. DDL (Data Definition Language)
  - 2. DML (Data Manipulation Language)
  - 3. TCL (Transaction Control Language)
  - 4. DCL (Data Control Language)
  - 5. DQL (Data Query Language)


SQL Commands
DDL
DCL
DML
TCL
DQL
Create
Drop
Alter
Truncate
Grant
Revoke
Insert
Update
Delete
Commit
Rollback
Save Point
Select

DDL – Data Definition Language

Data Definition Language actually consists of the SQL commands that can be used to define the database schema.

It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

Let’s understand all the keywords in DDL.

CREATE

- Create keyword can be used to create different database objects like tables, indexes, views etc.
- A table in databases is used to store data in a tabular format i.e. in form of rows and columns. It is similar like spreadsheet.
- A table creation command requires three things i.e. Name of Table, Name of fields (columns) in table, Data Type for each column.
  - Syntax:
    - CREATE TABLE TABLE_NAME (
      COLUMN1 DATATYPE,
      COLUMN2 DATATYPE);

ALTER(Adding column to a table)

1. Alter command goes in combination with add, rename, modify and drop. We will look at the add scenarios first.

2. Suppose, you want to add a column or multiple columns to an existing table, delete a column or multiple columns in a existing table, rename a column or multiple columns in an existing table.

e.g. Let's suppose you have students table and you want to add a column age in that table so you can use the below syntax

```
Alter table table_name add column column_name datatype;
Alter table students add column age int unsigned;
```

Now, let's suppose you want to add height, weight and scores in the same table so you can use the below syntax.

```
Alter table table_name add column column_name datatype, add column column_name datatype..
Alter table students add column height decimal (5,1), add column weight decimal (5,2), add column score int unsigned;
```

DO NOT WORRY WE WILL BE DISCUSSING DATATYPES IN THE FURTHER SESSIONS.

3. Adding columns in the above manner will add the column at the end of the table.

4. If in case you want to add the column at a specific position you can use the below syntax:

```
Alter table table_name add column column_name data type after column_name;
```

e.g. You want to add a column as Date of Birth next to the age column in the students table so the syntax will be

```
Alter table students add column date_of_birth date after age;
```

NOTE: There is nothing called as before however, if you would like to add a column at the beginning it means that it should be the first column in your table you can use the below syntax:

```
Alter table table_name add column column_name datatype first;
```

e.g. Let's supposed you want to add a new column student_id and that should be the first column in your table you can use the below syntax:

```
Alter table students add column student_id int unsigned first;
```

ALTER(Renaming a column in a table)

1. You can use the rename along with the ALTER keyword whenever you would like to rename a column

1. In the last slide you added a column named as student_id at the first position in the table. Let's suppose you want to rename this column just to ID, so you can use the below syntax:

    Alter table table_name rename column old_ column_name to new_column_name;

    Alter table students rename column student_id to id;

3. If in case you want to rename multiple columns in one single query. Suppose you want to change the height, weight and score column names to something the columns that we added in the last slide.

    Alter table table_name rename old_column_name to new_column_name, rename old_column_name to new_column_name;

    Alter table students rename column height to Tall, rename column weight to heavy, rename column score to points.

ALTER(Modifying column properties)

1. Modify keyword is also used along with the ALTER statement and it is useful for changing the data type of the column (logically if it is possible) or increasing the space or size of a column.

1. Consider the below example there is a column named a student_id and it was stored in text in the students table and now we want to convert the same in the integer format.

Student_id
1
2
3
4
5

You, can use the below syntax for the same.

Alter table table_name modify column column_name datatype;
Alter table students modify column student_id int unsigned;

Alter table table_name modify column column_name varchar(30)

The size of the column was Varchar(20). Using this query I changed the size of the column to Varchar(30).

ALTER(Dropping Column from a table)

1. Consider the below students table.

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

1. Let's suppose in this table you don't want the weight column in your table i.e. you want to delete this column. So, you can use the DROP keyword in the ALTER statement in the below manner .

   Alter table table_name drop column column_name;
   Alter table students drop column weight;

1. If, you want to drop two or three columns together in one statement you can use the below syntax.

   Alter table table_name drop column column_name, drop column_column_name;
   Alter table students drop column Name, drop column age;

DROP TABLE OR DATABASE

1. Consider the below students table.

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

1. Let's suppose the above table students is of no use now as it contains only redundant data and you would like to delete this table from the database you can write the below query!

```
DROP TABLE TABLE_NAME
DROP Table Students;
```

Remember, DROP is a DDL statement which is on auto save it means if you drop the table or database like this it is permanently gone.

1. If in case at any point of time you would like to get rid of the complete database you will write the below query.

```
DROP Database Database_Name;
```

TRUNCATE

1. Truncate statement empties the entire table.

1. Let's suppose you have students table which has 10,000 rows of data in it. If you write a TRUNCATE statement on the table all the rows (10,000) will be deleted in one go.

1. Consider the below table i.e. students

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

1. As you can see in the above table there are five rows if I write the below query all the five rows will be deleted in one go.

```
Truncate table table_name;
Truncate Table students;
```

COMMENT

1. Sometimes in tables we have some columns which are not easy to understand so we can leave a comment on the column so that it is easy for the end user to understand the real meaning of such columns.

1. Consider the below table students. We can understand all the columns but we are not sure what this bpm column means. So, what we can do is we can leave a comment on the column.

Student_id	Name	Age	Weight	bpm
1	A	19	50	99
2	B	22	55	78
3	C	25	67	89
4	D	19	51	88
5	E	22	59	95

2. We can use the below syntax to put a comment on the column.
   Alter table table_name modify column column name datatype comment " Your comments"
   Alter table students modify column bpm int unsigned comment " Heart Beat for the student".

1. In order to check the comments on a column we can write the below query:
   Show full columns from table_name;
   Show full columns from students;

DML & TCL

- Data Manipulation Language helps us to work on the exact data points i.e. it helps us to work with the rows in a table as rows in a table consist of a exact data points.

- Whenever, you make changes to the data in the table you need to confirm that whether you would like to save the changes that you have made or you would like to undo those changes.

- For this reason Data Manipulation Language and Transaction Control Language goes hand in hand as DML will help you to make changes to the data values and TCL will help you to save or undo the changes that you have made.
- Every time you write a DML statement a transaction gets initiated in the background and you need to write TCL statements to close that transaction.

DML Keywords – (INSERT, UPDATE, DELETE)

INSERT

1. Insert keyword or statement is used whenever we want to insert values in a table.

1. Consider this example. I have a table named students as shown in the screenshot below.

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

2. I want to add a new row for the student named F so I will use the below syntax.

Insert into table_name values (col1value,col2value,col3value,col4value)

Insert into students values (6,'F',22,79);

Note: Make sure the values you are mentioning it should be in the same sequence as per the column name as the first value will go to column1, second value will go to column2 and so on.

4. If in case you know the values only for let's say student_id and Weight and you want to insert values only in these two columns (condition that there is no not null constraint on any column) you can use the below query

Insert into students (student_id,weight) values (6,70);

• Once you have inserted the data, only you and only you can see the data in the table. If there are other people who have access to the same table and they try to see the data that you have inserted they will not be able to see anything as INSERT is a DML statement and every DML statement needs to be saved in order to make the changes permanent in the table.

→ In order to save the DML statement like in this scenario the INSERT statement you will write the below TCL command and execute it which acts like CTRL+S of Excel.

```
COMMIT;
```

→ Let's suppose you have inserted some rows into a table and later you realized that you would not like to go ahead in saving those records so you can write the below statements which acts like CTRL+Z of Excel.

```
ROLLBACK;
```

DML Keywords – (INSERT, UPDATE, DELETE)

DELETE

1. Delete keyword or statement is used to delete the values in a table.

1. Consider this example. I have a table named students as shown in the screenshot below.

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

2. I want to delete the student whose name is A

Delete from table_name where condition

```
Delete from Students where student_id = 1;
```

Note: Make sure you are giving some unique identifier in the where clause for system to understand which row to delete otherwise system will delete all the rows i.e. the entire table will go empty all the rows will be deleted.

- Once you have deleted the data, only you and only you can see the deleted data in the table. If there are other people who have access to the same table and they try to see the data they will still see the data as DELETE is a DML statement and every DML statement needs to be saved in order to make the changes permanent in the table.

→ In order to save the DML statement like in this scenario the Delete statement you will write the below TCL command and execute it which acts like CTRL+S of Excel.

```
COMMIT;
```

→ Let's suppose you have deleted some rows into a table and later you realized that you would not like to go ahead in saving those records so you can write the below statements which acts like CTRL+Z of Excel.

```
ROLLBACK;
```

DML Keywords – (INSERT, UPDATE, DELETE)

UPDATE

1. Update keyword or statement is used update the values in a table.

1. Consider this example. I have a table named students as shown in the screenshot below.

Student_id	Name	Age	Weight
1	A	19	50
2	B	22	55
3	C	25	67
4	D	19	51
5	E	22	59

2. I want to change the weight of the student whose name is A from 50 to 70

```
Update table_name set column_name  = newvalue where condition;
Update students set weight = 70 where student_id = 1;
```

Note: Make sure you are giving some unique identifier in the where clause for system to understand which row to update otherwise system will update all the rows for that column.

- Once you have updated the data, only you and only you can see the updated data in the table. If there are other people who have access to the same table and they try to see the data they will still see the old data as UPDATE is a DML statement and every DML statement needs to be saved in order to make the changes permanent in the table.

→ In order to save the DML statement like in this scenario the Update statement you will write the below TCL command and execute it which acts like CTRL+S of Excel.

```
COMMIT;
```

→ Let's suppose you have updated some rows into a table and later you realized that you would not like to go ahead in saving those records so you can write the below statements which acts like CTRL+Z of Excel.

```
ROLLBACK;
```

DQL (Data Query Language)

- DQL consist of one keyword called as SELECT.
- Whenever we want to query the data from the table in a database we will make use of the SELECT statement.
- If you would like to see all the data in a table in a database you can write the below query (* represents all) which means you will be able to see all the rows and columns in your data.
- If you want to see only limited columns you need to write all the column names separated by comma in front of select statement.

Synatx :

```
Select * from table_name;
Select col1, col2,col3 from table_name;
```

DCL (Data Control Language)

- DCL languages are used to control the user access to database, tables and other objects in the databases.
- DCL consist of two keywords
  - GRANT – Whenever, we need to provide access to the user for database or its objects we make use of GRANT statement.
  - REVOKE – Whenever, we need to withdraw the access of a user from the database or its objects we make use of REVOKE statement.

MySQL Data Types

- MySQL supports a wide variety of data types. Some of the most commonly used data types are listed below.
    - NUMERIC
    - DATE
    - DATETIME
    - TIMESTAMP
    - STRING

Type	Length in Bytes	Minimum Value (Signed)	Maximum Value (Signed)	Minimum Value (Unsigned)	Maximum Value (Unsigned)
TINYINT	1	-128	127	0	255
SMALLINT	2	-32768	32767	0	65535
MEDIUMINT	3	-8388608	8388607 to	0	16777215
INT	4	-2147483648	2147483647	0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807	0	18446744073709551615

INTEGER TYPES

Type	Length in Bytes	Minimum Value (Signed)	Maximum Value (Signed)	Minimum Value (Unsigned)	Maximum Value (Unsigned)
FLOAT	4	-3.402823466E+38	-1.175494351E-38	1.175494351E-38	3.402823466E+38
DOUBLE	8	-1.7976931348623 157E+ 308	-2.225073858507201 4E- 308	0, and 2.225073858507201 4E- 308	1.797693134862315 7E+ 308

FLOATING-POINT TYPES

Fixed – Point Types

- There is another data type as DECIMAL which is written as DECIMAL (10,3) where 10 is called as the precision and 3 is the scale. It means we will be able to store any value with 10 digits and 3 decimals.

- The maximum number of digits for DECIMAL is 65, but the actual range for a given DECIMAL column can be constrained by the precision or scale of a given column.

DATETIME, DATE and TIMESTAMP

Types	Description	Display Format	Range
DATETIME	Use when you need values containing both date and time information.	YYYY-MM-DD HH:MM:SS	'1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
DATE	Use when you need only date information.	YYYY-MM-DD	'1000-01-01' to '9999-12-31'.
TIMESTAMP	Values are converted from the current timezone to UTC while storing, and converted back from UTC to the current time zone when retrieved.	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC

TIME TYPE

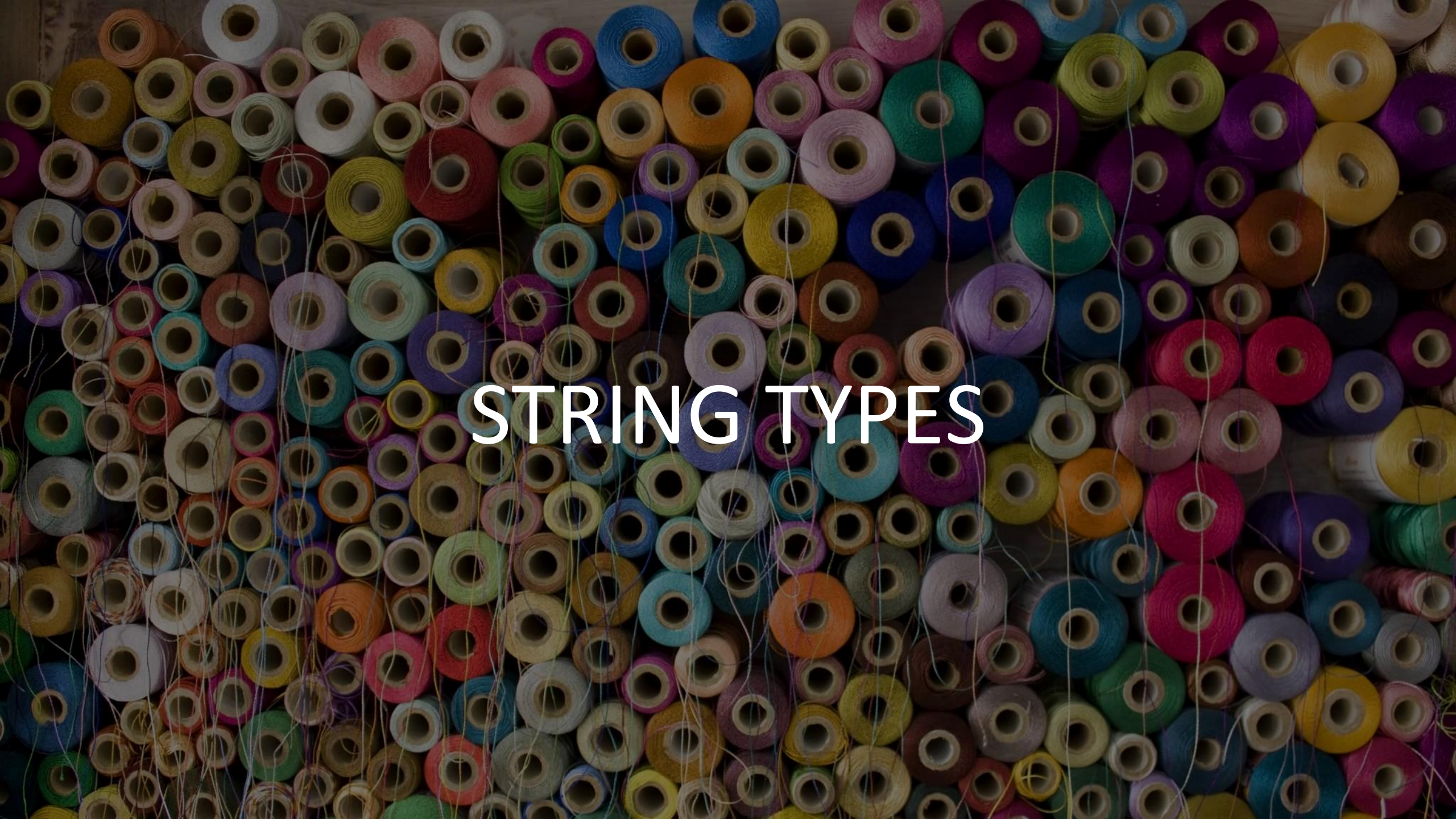
► *MySQL fetches and displays TIME values in 'HH:MM:SS' format or 'HHH:MM:SS' format The range of. TIME values from '-838:59:59' to '838:59:59'. The hours part may be rather large because not only the TIME type can be used to represent the time of day, i.e. less than 24 hours, but also the passed time or a time of interval between two events.*

► *The TIME values in MySQL can be recognized in different formats, some of which can include a trailing fractional seconds part in up to 6 digits microseconds precision. The range for TIME values is '-838:59:59.000000' to '838:59:59.000000'.*

YEAR TYPE

► *The YEAR type is a 1-byte type used to represent year values. It can be declared as YEAR(2) or YEAR(4) to specify a display width of two or four characters. If no width is given the default is four characters*

String length	Range
4-digit string	'1901' to '2155'.
4-digit number	1901 to 2155.
1- or 2-digit string	'0' to '99'. Values in the ranges '0' to '69' and '70' to '99' are converted to YEAR values in the ranges 2000 to 2069 and 1970 to 1999.
1- or 2-digit number	1 to 99. Values in the ranges 1 to 69 and 70 to 99 are converted to YEAR values in the ranges 2001 to 2069 and 1970 to 1999.


STRING TYPES

CHAR and VARCHAR Types

► *The CHAR and VARCHAR types are similar, but differ in the way they are stored and retrieved. They also differ in maximum length and in whether trailing spaces are retained.*

Types	Description	Display Format	Range in characters
CHAR	Contains non-binary strings. Length is fixed as you declare while creating a table. When stored, they are right-padded with spaces to the specified length.	Trailing spaces are removed.	The length can be any value from 0 to 255.
VARCHAR	Contains non-binary strings. Columns are variable-length strings.	As stored.	A value from 0 to 255 before MySQL 5.0.3, and 0 to 65,535 in 5.0.3 and later versions.