19CSE202 - Database Management System

LAB 2 - BASIC SQL QUERIES

Vidhya. S
Assistant Professor(Sr. Gr)
Department of Computer Science & Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Coimbatore

Overview of SQL Query Language

- SQL Data Definition
  - Basic Types
  - Basic Schema Definition
  - Queries on a Single Relation
- Modification of the Databases
  - Deletion
  - Insertion
  - Updates

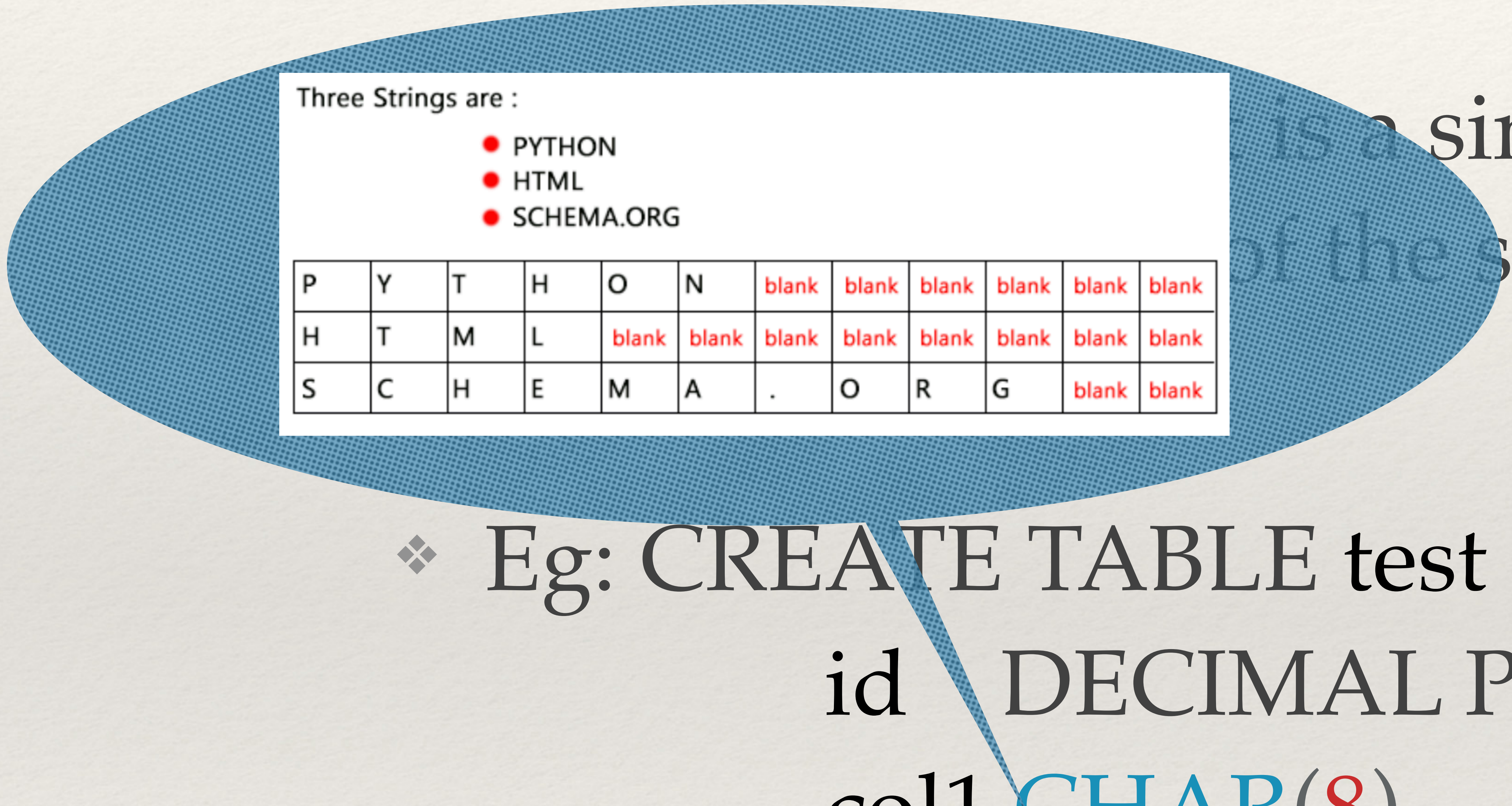
Basic Data Types

- Character String Types: CHARACTER, VARCHAR
- Boolean Type: Stores truth values - either TRUE or FALSE.
- Numeric Types: INTEGER, DECIMAL(p, s), NUMERIC(p, s), FLOAT(p), REAL, SMALLINT, BIGINT
- Datetime Types: DATE - Represents a date. Format : yyyy-mm-dd

Character String Types

- CHARACTER: Character string, fixed length.

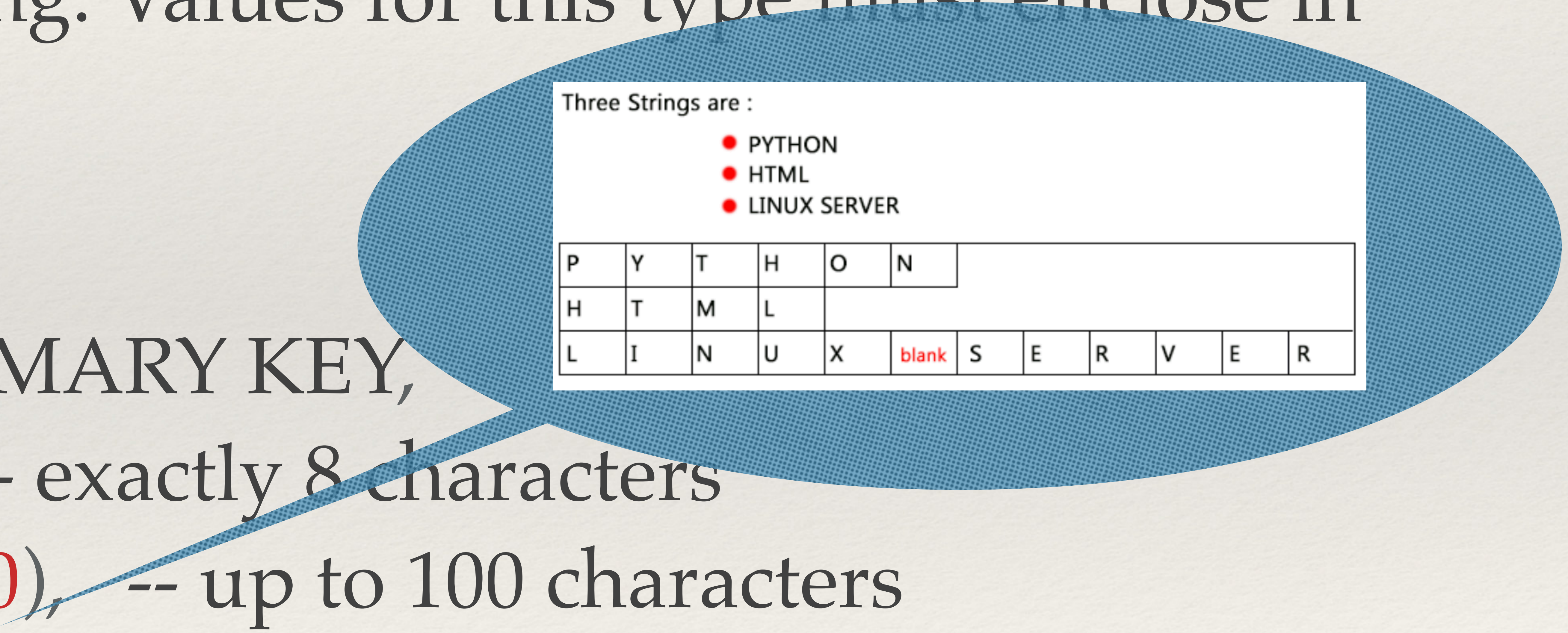
is a single nonnegative integer that refers to the of the string. Values for this type must enclose in



Three Strings are :

- PYTHON
- HTML
- SCHEMA.ORG

P	Y	T	H	O	N	blank	blank	blank	blank	blank	blank
H	T	M	L	blank	blank	blank	blank	blank	blank	blank	blank
S	C	H	E	M	A	.	O	R	G	blank	blank



Three Strings are :

- PYTHON
- HTML
- LINUX SERVER

P	Y	T	H	O	N						
H	T	M	L								
L	I	N	U	X	blank	S	E	R	V	E	R

- Eg: CREATE TABLE test (

```
        id   DECIMAL PRIMARY KEY,
        col1 CHAR(8),        -- exactly 8 characters
        col2 VARCHAR(100),   -- up to 100 characters
    );
```

NUMERIC DATA TYPE

Specified As	Stored As
DECIMAL(p, s)	Exact numerical, precision p, scale s. A decimal number, that is a number that can have a decimal point in it. The size argument has two parts : precision and scale. The scale can not exceed the precision. Precision comes first, and a comma must separate from the scale argument.
NUMERIC(p, s)	Exact numerical, precision p, scale s.
FLOAT(p)	Approximate numerical, mantissa precision p. Precision is greater than or equal to 1 and the maximum precision depends on the DBMS.
REAL	Same as FLOAT type except that the DBMS defines the precision.

NUMERIC

- Example : Precision and Scale Examples for 235.89

Specified As	Stored As
NUMERIC(5)	236
NUMERIC(5, 0)	236
NUMERIC(5, 1)	235.9
NUMERIC(5, 2)	235.89
NUMERIC(4,0)	236
NUMERIC(4,1)	235.9
NUMERIC(4.2)	Exceed Precision
NUMERIC(2,0)	Exceed Precision

NUMERIC - Eg

```
CREATE TABLE test (
         id    DECIMAL PRIMARY KEY,
        name  VARCHAR(100),   -- up to 100 characters
        col1  DECIMAL(5,2),    -- three digits before the decimal and two behind
        col2  SMALLINT,         -- no decimal point
        col3  INTEGER,          -- no decimal point
        col5  FLOAT(2),         -- two or more digits after the decimal place
        col6  REAL,
        col7  DOUBLE PRECISION
);
```

DDL

- CREATE: Create an object. I mean, create a database, table, triggers, index, functions, stored procedures, etc.
- ALTER: Used to alter the existing database or its object structures.
- DROP: Command helps to delete objects. For example, delete tables, delete a database, etc
- TRUNCATE: This SQL DDL command removes records from tables. Including all spaces allocated for the records are removed.
- RENAME: Renaming the database objects.

DDL - CREATE

- Syntax:

```
CREATE TABLE table_name(
        column1 datatype,
        column2 datatype,
        column3 datatype,
        …..
        columnN datatype,
        PRIMARY KEY( one or more columns )
);
```

Eg: CREATE TABLE Persons (

```
  PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

Possible Errors :

Error 1:: Table already exist

Sol: Table name already exist, create new_table name

Error 2 : Table or view does not exist
(table is not existing
or created in the server)

Sol: create new table

DDL - ALTER

- Adds a column to a table. If not specified otherwise, the column will be added at the end of the table.
- Syntax:

-
  ```
  ALTER TABLE table
  ADD [COLUMN] column_name column_definition [FIRST|AFTER
    existing_column];
  ```

Eg: **ALTER TABLE** vendors
```
ADD COLUMN phone VARCHAR(15) AFTER name;
```

- Drop column name

Eg: **ALTER TABLE** Customers
DROP COLUMN ContactName;

- Modify the data type of the column name

Eg: **ALTER TABLE** *table_name*
MODIFY COLUMN column_name datatype;

DDL - DROP

- DROP TABLE *table_name;*
- TRUNCATE TABLE *table_name;*
  Difference between DROP AND TRUNCATE ?
  - DROP – Delete the schema
  - TRUNCATE – Schema exist , all the records will be delete

DDL - RENAME

- RENAME TABLE old_tablename TO new_tablename

(OR)

- ALTER TABLE old_tablename RENAME TO new_tablename

DML - INSERT

- INSERT INTO table_name (column1, column2, column3,...columnN) VALUES (value1, value2, value3,…valueN);

(OR)

Eg: INSERT INTO Customers (CustomerName, City, Country)
VALUES ('Cardinal', 'Stavanger', 'Norway');

- INSERT INTO table_name VALUES (value1, value2, value3,…valueN);

(OR)

Eg : INSERT INTO Customers (CustomerName, City, Country)
VALUES ('Cardinal', 'Stavanger', 'Norway');

- INSERT INTO table_name VALUES (&column1, &column2, &column3,...&columnN)

Eg: INSERT INTO Customers VALUES
(‘&CustomerName’, ‘&City’, ‘&Country’)

DML - UPDATE

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

- Eg: UPDATE Customers
  SET ContactName = 'Alfred', City= 'Frankfurt'
  WHERE CustomerID = 1;

DML - DELETE

- DELETE FROM *table_name* WHERE *condition;*
  Eg:
  DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';