

```
1 CDIP-UIU: Python & Data Analysis
2 Batch - 09
3
4 Welcome to the 1st lecture on MySQL!
5
6
7 Text Books (optional)
8 -----
9 1. Database System Concepts by Abraham Silberschatz, Henry F. Korth and S.Sudarshan
10 2. Database Systems: The Complete Book, by Garcia-Molina, Ullman and Widom
11
12
13 References
14 -----
15 1. https://dev.mysql.com/doc/refman/8.0/en/
16 2. http://www.mysqltutorial.org/
17 3. https://www.w3resource.com/sql-exercises/
18 4. https://www.w3schools.com/mysql/default.asp
19 5. https://www.coffeendcode.com/MYSQL/introduction.htm
20
21
22 =====
23
24
25 Database - Collection of correlated data
26 DBMS - Organization/Management/Retrieval of data from database
27
28 Database system = Database + DBMS (Database Management System)
29
30 Website ---> MySQL(high level) ---> DBMS ---> database(low level)
31 Web server Database server
32
33 PC <==> Internet <==> Web server <==> Database server
34
35 SQL (standard) SQL --> Structured Query Language
36 - MySQL
37 - Oracle
38 - MS SQL
39 - PostgreSQL
40
41 Web server - Apache
42 DB server - MySQL
43
44 XAMPP (Bundle) - has apache and MySQL built-in
45
46 1. Open XAMPP and turn on Apache and MySQL server
47 2. Go to browser and type any of the following
48 - http://localhost/phpmyadmin/
49 - http://127.0.0.1/phpmyadmin/
50 - http://127.0.0.1:[port_number]/phpmyadmin/
51 (Example: http://127.0.0.1:80/phpmyadmin/)
52
53
54
55 MySQL - case-insensitive, keyword Uppercase
56
57 MySQL - DDL -- Data definition language
58 - create/drop database
59 - create/drop/alter table
```

```

60
61     - DML -- Data manipulation language
62     - Insert data
63     - Delete data
64     - Search data / SELECT
65     - Update data
66
67 File extension for MySQL files: .sql
68
69 =====
70
71 -----
72 | Data Definition Language |
73 -----
74
75 Database:
76 -----
77
78 How to create a database:
79 - CREATE DATABASE [database_name]
80 - CREATE DATABASE IF NOT EXISTS [database_name]
81
82 How to delete/drop a database:
83 - DROP DATABASE [database_name]
84 - DROP DATABASE IF EXISTS [database_name]
85
86
87 Table:
88 -----
89
90 Databases are a set of tables. May contain one or more tables.
91
92 Example student table
93 -----
94
95 -----
96 | name | dob      | admission_date | cgpa |
97 -----
98 | A    | 01.01.95 | 01.01.15      | 3.50 |
99 -----
100 | B    | 04.02.95 | 02.01.15      | 3.33 |
101 -----
102
103
104 Python class example
105 -----
106
107 class Student:
108     def __init__(self, name, dob, admission_date, cgpa):
109         self.name = name
110         self.dob = dob
111         self.admission_date = admission_date
112         self.cgpa = cgpa
113
114 student_1 = new Student('A', '01.01.95', '01.01.15', 3.50)
115 student_2 = new Student('B', '04.02.95', '02.01.15', 3.33)
116
117
118 How to create a table in a database:
119 - CREATE TABLE table_name (

```

```

120     column1 datatype,
121     column2 datatype,
122     column3 datatype,
123     ....
124 );
125
126 Python datatypes:
127 - int      [numeric]
128 - float    [numeric]
129 - string   [text]
130 - boolean  [boolean - true/false]
131 - complex  [numeric] (a + ib)
132
133 name = "Shabab" [compiler would automatically understand that 'name' is a variable
134 and its datatype is string, looking at the datatype]
135
136 MySQL datatypes:
137 - INT (size) -- by default 11
138 - FLOAT (m, d) --> Example: 3.1416; m = 5, d = 4
139   FLOAT (10, 5) --> xx, xxx . xxxxx
140 - DOUBLE (m, d)
141
142 - VARCHAR (size) --> Variable characters
143   VARCHAR(30) --> "This is a boy"
144
145 - DATE -- format: 'YYYY-MM-DD' = '1995-01-01'
146 - DATETIME -- format: 'YYYY-MM-DD hh:mm:ss' = '1995-01-01 23:59:59'
147 - TIME -- format: 'hhh:mm:ss' = '15:07:23' >> Range: '-838:59:59' to '838:59:59'
148
149 Table creation example:
150 -----
151
152 CREATE TABLE students (
153     name VARCHAR(50),
154     dob DATE,
155     admission_date DATE,
156     cgpa FLOAT(3, 2),
157     courses INT
158 );
159
160 (Without using underscore. Using ``)
161
162 CREATE TABLE teachers (
163     id int(10),
164     name VARCHAR(50),
165     `date of birth` DATE,
166     `joining date` DATE
167 );
168
169
170
171 How to delete/drop a table:
172 - DROP TABLE [table_name]
173 - DROP TABLE IF EXISTS [table_name]
174
175
176 Alter table:
177 -----
178

```

```

179 How to alter a table:
180     Alter --> change/edit
181
182 - add column
183 - delete column
184 - datatype changing
185
186 - column rename
187 - table rename
188
189
190 Alter table - add column:
191 -----
192
193 ALTER TABLE table_name
194 ADD column_name datatype;
195
196 ALTER TABLE students
197 ADD department VARCHAR(50);
198
199 ALTER TABLE students
200 ADD id INT(10);
201
202
203 Alter table - drop column:
204 -----
205
206 ALTER TABLE table_name
207 DROP COLUMN column_name;
208
209 ALTER TABLE students
210 DROP COLUMN admission_date;
211
212
213 Alter table - change column datatype:
214 -----
215
216 ALTER TABLE table_name
217 MODIFY COLUMN column_name datatype;
218
219 ALTER TABLE students
220 MODIFY COLUMN department INT(15);
221
222
223 MySQL Constraints:
224 -----
225 - NOT NULL - Ensures that a column cannot have a NULL value
226 - UNIQUE - Ensures that all values in a column are different
227 - DEFAULT - Sets a default value for a column if no value is specified
228
229 CREATE TABLE students (
230     id INT,
231     name VARCHAR(30) UNIQUE,
232     dob DATE,
233     admission_date DATE,
234     cgpa FLOAT(3,2) DEFAULT 0.00,
235     dept VARCHAR(5) NOT NULL
236 );
237
238 (Example of NOT NULL, Unique and DEFAULT)

```

```

239
240 -----
241 | ID | name | dob      | admission_date | cgpa | dept |
242 -----
243 | 1  | A    | 01.01.95 | 01.01.15      | 3.50 | CSE  |
244 -----
245 | 2  | B    | 04.02.95 | 02.01.15      | 3.33 | NULL | (Error --> NULL in dept)
246 -----
247 | 3  | C    | 18.02.95 | 01.01.15      | 3.63 | BBA  |
248 -----
249 | 4  | A    | 01.04.95 | 01.01.15      | 3.46 | BBA  | (Error --> unique name)
250 -----
251 | 5  | E    | 08.05.96 | 01.01.16      | 0.00 | Civil | (Default --> cgpa = 0.00)
252 -----

```

```

253
254
255 - PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row
in a table
256 - FOREIGN KEY - Prevents actions that would destroy links between tables
257
258

```

259 What is a primary key?

260 - A unique identifier for a table's rows.

261

262 Students table:

263 =====

264

```

265 -----
266 | ID (pk) | name | dob      | cgpa | dept (fk) |
267 -----
268 | 1       | A    | 01.01.95 | 3.50 | 1         | --> a student's info
269 -----
270 | 2       | B    | 04.02.95 | 3.33 | 2         |
271 -----
272 | 3       | C    | 18.02.95 | 3.63 | 3         |
273 -----
274 | 4       | A    | 01.04.95 | 3.46 | 3         |
275 -----

```

276

277 What is a foreign key?

278 - A unique identifier that uniquely identifies an object/row/instance of another table

279

280 Departments table:

281 =====

282

```

283 -----
284 | ID (pk) | name   | head  | no_of_faculties |
285 -----
286 | 1       | CSE    | Dr. A | 20              |
287 -----
288 | 2       | EEE    | Dr. B | 18              |
289 -----
290 | 3       | BBA    | Dr. C | 22              |
291 -----
292 | 4       | Civil  | Dr. D | 23              |
293 -----

```

294

295

296 How to add primary key:

```

297 -----
298 - Using 'PRIMARY KEY' keyword (Also example of default & unique)
299 CREATE TABLE students (
300     id int PRIMARY KEY,
301     name VARCHAR(50) UNIQUE,
302     dob DATE,
303     admission_date DATE,
304     cgpa FLOAT(3,2) DEFAULT 0.00
305 );
306
307 - Using constraints
308 CREATE TABLE students (
309     id int,
310     name VARCHAR(50) UNIQUE,
311     dob DATE,
312     admission_date DATE,
313     cgpa FLOAT(3,2) DEFAULT 0.00,
314
315     CONSTRAINT pk_students PRIMARY KEY (id)
316 );
317
318
319 CANDIDATE KEY
320 -----
321 - Unique identifier
322 - Primary key is selected from candidate keys
323
324
325 Example of NOT NULL:
326 -----
327 CREATE TABLE departments (
328     id int PRIMARY KEY,
329     name VARCHAR(10),
330     head VARCHAR(50) NOT NULL,
331     no_of_faculties int
332 );
333
334
335 How to add foreign key:
336 -----
337
338 Note:
339 When we want to refer Table B in Table A,
340 Foreign key datatype of Table A = Primary key datatype of Table B --> (MUST BE SAME)
341
342 - While altering table
343 ALTER TABLE students
344 ADD dept int,
345 ADD CONSTRAINT fk_students_department FOREIGN KEY (dept) REFERENCES departments(id)
346
347
348 - While creating table
349 CREATE TABLE students (
350     id int AUTO_INCREMENT,
351     name VARCHAR(50) UNIQUE,
352     dob DATE,
353     admission_date DATE,
354     cgpa FLOAT(3,2) DEFAULT 0.00,
355     dept int,
356

```

```

357     CONSTRAINT pk_students PRIMARY KEY (id),
358     CONSTRAINT fk_students_department FOREIGN KEY (dept) REFERENCES departments(id)
359 );
360
361
362 =====
363
364
365 -----
366 | Data Manipulation Language |
367 -----
368
369
370 How to insert data in table:
371
372 - INSERT INTO [table_name] (col1, col2, col3., ...)
373   VALUES (val1, val2, val3, ...)
374
375 INSERT INTO `students`(`id`, `name`, `dob`, `admission_date`, `cgpa`, `dept`)
376 VALUES (1, 'A', '1995-01-01', '2015-01-01', '3.50', 1)
377
378
379 How to select data from table: (Show data)
380
381 SELECT col1, col2, ...,
382 FROM table_name,
383 WHERE condition
384
385 Example:
386 - SELECT name
387   FROM students
388
389 - SELECT name, cgpa
390   FROM students
391
392 How to select all columns: Using (*)
393 -----
394 - SELECT *
395   FROM students
396
397
398 Example of WHERE:
399 - SELECT name, cgpa
400   FROM students
401  WHERE id = 1
402
403 - SELECT name, dob
404   FROM students
405  WHERE cgpa = 0.00
406
407
408 Multiple conditions:
409 -----
410
411 When you have 2 or more conditions in WHERE clause.
412
413 - AND = Must satisfy both/all conditions
414 - OR = Must satisfy at least one condition
415
416

```

417 Example students table:

418 -----

419 -----

	ID	name	dob	admission_date	cgpa	dept
420						
421						
422	1	A	1995-01-01	2015-01-01	3.50	1
423						
424	2	B	1995-01-01	2015-01-01	0.00	1
425						
426	3	C	1995-01-01	2015-01-01	0.00	1
427						

427 -----

428

429 conditions:

430 -----

431 - SELECT name, dob
432 FROM students
433 WHERE cgpa = 0.00 AND id = 2

434

435

436

437 - SELECT name, dob
438 FROM students
439 WHERE cgpa = 0.00 OR id = 2

440

441

442

Output:

430 -----

	name	dob
431		
432		
433	B	1995-01-01
434		

434 -----

	name	dob
437		
438		
439	B	1995-01-01
440		
441	C	1995-01-01
442		