

Simple Queries in SQL

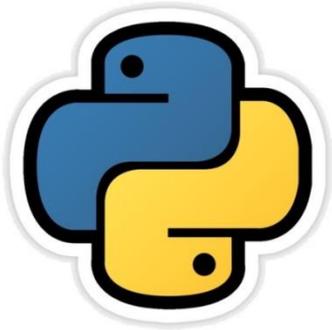
&

Table Creation and Data Manipulation

सीबीएसई पाठ्यक्रम पर आधारित

कक्षा -11

अध्याय – 17 & 18



द्वारा:

संजीव भदौरिया

स्नातकोत्तर शिक्षक (संगणक विज्ञान)

के० वि० बाराबंकी (लखनऊ संभाग)

परिचय

- SQL को 1970 के आस पास IBM लैब में विकसित किया गया था | इसे SEQUEL भी कहा जाता था जोकि एक non-procedural language थी |
- इसमें हमेशा यह बताया जाता है कि “क्या चाहिए” (WHAT is required?).
- SQL में हम निम्न कार्य कर सकते हैं -
 - I. *Database के structure को create और modify*
 - II. *System के लिए security settings में बदलाव*
 - III. *User को database या table के साथ कार्य करने की अनुमति देना*
 - IV. *Database से query करना*
 - V. *Database के डाटा को insert/modify/delete करना . . . इत्यादि |*
- इस अध्याय में हम table बनाना उनमें सुधार करना तथा साधारण query बनाना सीखेंगे |

कुछ MySQL के तत्व (Elements)

- MySQL के कुछ प्रमुख elements निम्न हैं –
 - *Literals*
 - *Datatypes*
 - *Nulls*
 - *Comments*
- **Literals** : इन्हें सामान्यतया fix data value कहते हैं | ये character, numeric या text literal हो सकती है |
- **Data Types** : ये निम्न प्रकार के होते हैं -
 - (i) **Numeric** : INT, TINYINT, SMALLINT, MEDIUMINT, FLOAT , DOUBLE इत्यादि |
 - (ii) **Date and Time** : DATE, DATETIME, TIME, YEAR इत्यादि |
 - (iii) **String type** : CHAR, VARCHAR, BLOB or TEXT TINYTEXT, ENUM इत्यादि |

कुछ MySQL के तत्व (Elements)

- **NULL Values** : जिस row के किसी column में कोई value न हो उसे NULL माना जा सकता है | NULL को किसी भी प्रकार के datatype के साथ रखा जा सकता है बशर्ते उस column को किसी बंधन में न रखा गया हो जैसे NOT NULL Constraint या primary key |
- **Comments** : एक Comment एक प्रकार text होता है जो कि execute नहीं होता है | comment का प्रयोग सिर्फ code को समझने के लिए लिखा जाता है | comment तीन प्रकार से दिया जा सकता है
 - `/* comment */`
 - `-- comment`
 - `# comment`

SQL COMMAND SYNTAX

- SQL पहले से परिभाषित commands का संकलन प्रदान करता है |
- SQL के syntax में निम्न तत्व होते हैं
 - Keyword:
 - Statements:
 - Clauses:
 - Arguments:
- SQL commands, case sensitive नहीं होते हैं | समस्त कमांड mysql> पर दिए जाते हैं |

Database बनाना

- Database बनाने के लिए निम्न कमांड का प्रयोग करते हैं –

```
mysql> CREATE DATABASE <database name >;
```

उदहारण के लिए -

```
mysql> create database school;
```

Database को प्रयोग करना

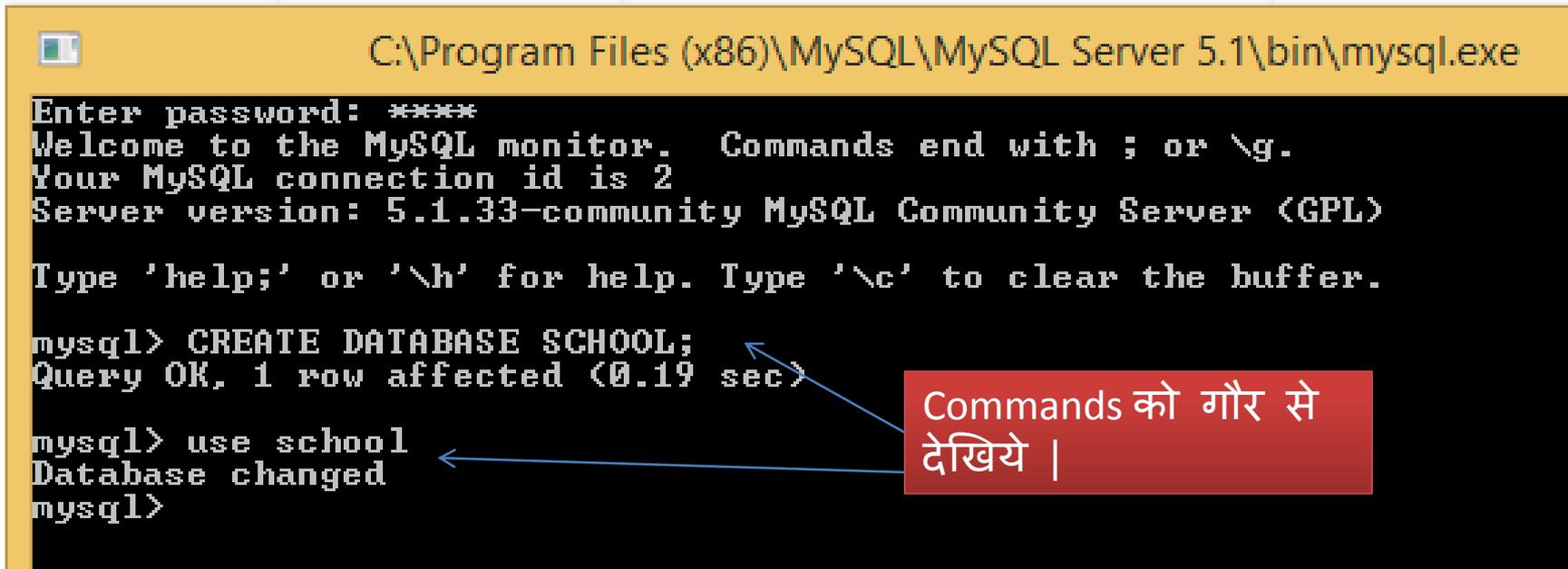
- Database को प्रयोग में लाने के लिए निम्न कमांड का प्रयोग करते हैं –

```
mysql> USE <database name >;
```

उदहारण के लिए -

```
mysql> USE school;
```

इसके पश्चात् एक सन्देश लिखा कर आयेगा “database changed”



```
C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.33-community MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE DATABASE SCHOOL;
Query OK, 1 row affected (0.19 sec)

mysql> use school
Database changed
mysql>
```

Commands को गौर से देखिये |

Table बनाना

- Database में table को बनाने के लिए निम्न कमांड का प्रयोग करते हैं –

```
mysql> CREATE TABLE <Table Name> (<Col1> <DataType(Size)>,  
                                     <Col2><DataType(size)>, . . . );
```

उदहारण के लिए -

```
mysql>create table student (Roll INT(4) Primary Key, Name CHAR(20),  
                             (Age INT(2), City CHAR(10) ) ;
```

इसके जवाब में एक सन्देश आता है “Query OK”

```
mysql> Create table Student(Roll INT(4) Primary key, Name CHAR(20),  
-> Age INT(2), City CHAR(10));  
Query OK, 0 rows affected (0.35 sec)
```

इसमें एक शब्द प्रयोग किया गया है “*primary key* “ जिसके बारे में आगे वाले पेज में चर्चा करेंगे | बस यहान्याह समझने के लिए काफी है की *primary key* लगाने से उस *column* में डाटा *unique* जायेगा तथा उसमे हम *duplicate value* नहीं डाल सकते |

Table का ढांचा देखना

- Database में table के ढांचे को देखने के लिए निम्न कमांड का प्रयोग करते हैं

```
mysql> DESC <TableName>;
```

उदहारण के लिए -

```
mysql>DESC Student;
```

इसके जवाब में table का पूरा ढांचा (structure) प्रदर्शित हो जाता है।

```
mysql> DESC Student;
```

Field	Type	Null	Key	Default	Extra
Roll	int(4)	NO	PRI	NULL	
Name	char(20)	YES		NULL	
Age	int(2)	YES		NULL	
City	char(10)	YES		NULL	

```
4 rows in set (0.00 sec)
```

Table के structure में परिवर्तन

- Database में table के ढांचे में परिवर्तन करने के लिए निम्न कमांड का प्रयोग करते हैं

```
mysql>ALTER TABLE <Table name> ADD/MODIFY  
(<Col> <type(size)>, . . . .)
```

उदहारण के लिए -

```
mysql> Alter Table Student Add (class INT(2));  
इसके जवाब में "Query OK" प्रदर्शित हो जाता है |
```

```
mysql> alter table student add (class INT(2));  
Query OK, 0 rows affected (0.42 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

फिर यदि आप DESC कमांड से table देखें तो

```
mysql> DESC Student;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type   | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Roll  | int(4) | NO   | PRI | NULL    |       |  
| Name  | char(20)| YES  |     | NULL    |       |  
| Age   | int(2)  | YES  |     | NULL    |       |  
| City  | char(10)| YES  |     | NULL    |       |  
| class | int(2)  | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.02 sec)
```

- एक नया column table के साथ जुड़ गया है |
- Column बढ़ाये जा सकते हैं |
- Column का आकार बदला जा सकता है |

Table को DROP करना

- Database में table को delete करने के लिए निम्न कमांड का प्रयोग करते हैं

```
mysql> DROP Table <Table Name>;
```

उदहारण के लिए -

```
mysql> drop table <Student>
```

इसके जवाब में एक सन्देश आता है “Query OK” और यदि table का structure देखना चाहें तो नहीं देख सकते क्योंकि table delete हो चुकी है |

```
mysql> drop table student;
Query OK, 0 rows affected (0.03 sec)

mysql> desc student;
ERROR 1146 (42S02): Table 'school.student' doesn't exist
mysql>
```

Table से सम्बंधित समस्त कमांड DDL की श्रेणी में आते हैं क्योंकि इनके द्वारा प्रयोग किये जाने वाले डाटा की परिभाषा दी जाती है |

Constraints के द्वारा Data Integrity

- किसी table के एक या अधिक column पर लगाये जाने वाले check या condition को **CONSTRAINTS** कहा जाता है ।
- चूँकि इन्हें integrity बरकरार रखने के लिए लगाया जाता है अतः इन्हें integrity constraints भी कहते हैं ।
- जैसे ही कोई constraint किसी table पर लगाया जाता है तो समस्त डाटा को उस नियम का पालन करना आवश्यक हो जाता है ।
- Constraints को table बनाते समय ही लगाया जाता है ताकि डाटा insert करते समय नियमों का पालन किया जा सके । इसके लिए syntax निम्न है -

```
mysql> CREATE TABLE <TableName>  
      (<Col1> <type(size)> <Column Constraint>,  
      <Col2> <type(Size)> <Column Constraint>, . . . . );
```

Table बनाने से पहले constraints के बारे में जान लेते हैं ।

Integrity Constraints

- Constraints किसी भी database की integrity को सुनिश्चित करते हैं | कुछ constraints निम्न हैं -
- **Unique Constraint** : यह constraint tab लगते हैं जब column में डाटा को unique रखना हो | यह NULL value को अनुमति देता है |
- **Primary key Constraint** : यह तब प्रयोग करते हैं जब डाटा को uniquely identify करना हो तथा अन्य table के साथ रिफरेन्स देना हो | यह NULL value को अनुमति नहीं देता |
- **Default Constraint** : यह तब प्रयोग करते हैं जब user किसी column के लिए कोई value न दे तो स्वतः कुछ न कुछ value table में insert हो जाये |
- **Check Constraint** : यह value को सीमा में बंधने के लिए प्रयोग में लाया जाता है जैसे age वाले column में 20 साल से नीचे insert न हो |
- **Foreign key Constraint** : किसी table में ऐसा non-key attribute जो किसी दूसरे table के primary key से व्युत्पन्न (Derived) किया गया हो उसे वर्तमान table में foreign key कहा जाता है | यह अधिकतर **transaction table** में होती है|

अब हम एक ऐसी table बनाते हैं जिसमे ये constraints प्रयोग किये जाएँ |

Table को बनाना

```
mysql> create table student
```

```
(Roll INT(4) PRIMARY KEY,  
NAME CHAR(20) NOT NULL,  
Age INT(2) CHECK (age>5),  
Class CHAR(3) Default 'I',  
City CHAR(10));
```

यह column primary key बनेगा |

इस column में null नहीं जायेगा |

इस column में 5 से कम age नहीं जाएगी|

कुछ भी न देने पर 'I' स्वतः insert हो जायेगा |

इस प्रकार लगाये गए सभी constraints column लेवल constraints कहलाते हैं |

```
mysql> CREATE TABLE Student  
-> (ROLL INT(4) PRIMARY KEY,  
-> NAME CHAR(20) NOT NULL,  
-> AGE INT(2) CHECK(AGE>5),  
-> CLASS CHAR(3) DEFAULT 'I',  
-> CITY CHAR(10));  
Query OK, 0 rows affected (0.09 sec)
```

Create table कमांड

```
mysql> DESC Student;
```

Field	Type	Null	Key	Default	Extra
ROLL	int(4)	NO	PRI	NULL	
NAME	char(20)	NO		NULL	
AGE	int(2)	YES		NULL	
CLASS	char(3)	YES		I	
CITY	char(10)	YES		NULL	

5 rows in set (0.00 sec)

DESC कमांड

Table level constraints बनाना

```
mysql> create table student
      (Enroll INT(4),
       Roll INT(4) ,
       NAME CHAR(20) NOT NULL,
       Age INT(2),
       Class CHAR(3) Default 'I',
       City CHAR(10),
       PRIMARY KEY (Enroll, Roll));
```



इस प्रकार लगाये गए सभी constraints Table लेवल constraints कहलाते हैं | इन्हें सामान्यतया तब लगते हैं जब constraints को columns के ग्रुप पर लगाना हो |

Table में Record insert करना

Table में record insert करने के लिए निम्न syntax होता है -

```
mysql> INSERT INTO <TableName> (<Col1> <Col2> <Col3> <Col4>  
VALUES (<val1>,<val2>,<val3>,<val4>,.. .);
```

```
mysql> INSERT INTO STUDENT (ROLL, NAME, AGE, CLASS, CITY) VALUES  
-> (1001,'Pankaj',6,'I','Barabanki');  
Query OK, 1 row affected (0.03 sec)
```

इसमें हम column के क्रम बदल सकते हैं |

```
mysql> INSERT INTO STUDENT (ROLL, NAME, CLASS,AGE, CITY) VALUES  
-> (1002,'Naresh','II',8,'Kanpur');  
Query OK, 1 row affected (0.01 sec)
```

इसमें हम बिना column के नाम दिए values insert कर सकते हैं बस ये ध्यान रखना होगा कि column का क्रम फिर table का ही होगा |

```
mysql> INSERT INTO STUDENT VALUES  
-> (1003,'Sunita',7,'II','Barabanki');  
Query OK, 1 row affected (0.03 sec)
```

Table में Record insert करना

Default value insert करने के लिए सिर्फ उन्ही column को specify करते हैं जिनमे value insert करनी हो | तो छोड़े गए column जिसमे default set किया गया होगा उसमे default value आयेगी तथा जिसमे default set नहीं किया गया होगा उसमे NULL जायेगा |

```
mysql> INSERT INTO STUDENT (ROLL, NAME, AGE) VALUES  
-> (1004, 'Meera', 9);  
Query OK, 1 row affected (0.05 sec)
```

Table के records को display करें तो -

```
mysql> Select * from Student;
```

ROLL	NAME	AGE	CLASS	CITY
1001	Pankaj	6	I	Barabanki
1002	Naresh	8	II	Kanpur
1003	Sunita	7	II	Barabanki
1004	Meera	9	I	NULL
1005	Nita	7	I	NULL

5 rows in set (0.00 sec)

City में कोई constraint नहीं लगाया तो NULL insert हो गया |

ये default set किये गए थे constraints के द्वारा

Table में Record Update करना

Table में record update करने के लिए निम्न syntax होता है -

```
mysql> UPDATE <TableName> SET <ColName>=<NewValue>  
WHERE <Condition>
```

```
mysql> Select * from Student;
```

ROLL	NAME	AGE	CLASS	CITY
1001	Pankaj	6	I	Barabanki
1002	Naresh	8	II	Kanpur
1003	Sunita	7	II	Barabanki
1004	Meera	9	I	NULL
1005	Nita	7	I	NULL

5 rows in set (0.00 sec)

इस table में Meera की age 6 तथा Meera और Nita की City में lucknow करना है | इसके लिए अलग अलग कमांड प्रयुक्त होंगे |

```
mysql> update student set age=6 where roll=1004;  
Query OK, 1 row affected (0.06 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

इस कमांड ने Meera की age बदली |

```
mysql> update student set city = 'Lucknow'  
-> where roll in (1004,1005);  
Query OK, 2 rows affected (0.03 sec)  
Rows matched: 2 Changed: 2 Warnings: 0
```

इस कमांड ने Meera व nita की City बदली |

Table से Record को Delete करना

Table से record को delete करने के लिए निम्न syntax होता है -

```
mysql> DELETE FROM <TableName> WHERE <Condition>
```

```
mysql> Delete from Student where roll in (1001,1004);  
Query OK, 2 rows affected (0.03 sec)
```

```
mysql> select * from Student;
```

ROLL	NAME	AGE	CLASS	CITY
1002	Naresh	8	II	Kanpur
1003	Sunita	7	II	Barabanki
1005	Nita	7	I	Lucknow

```
3 rows in set (0.00 sec)
```

Delete करने के बाद table के contents को देखने के लिए ।

Table के सारे records को delete करने के लिए निम्न कमांड चलते हैं ।

```
mysql> delete from student;  
Query OK, 3 rows affected (0.03 sec)
```

```
mysql> select * from student;  
Empty set (0.00 sec)
```

Table को access करना

Table से डाटा को access करने के लिए निम्न syntax होता है -

```
mysql> SELECT <Col Names> FROM <Table Name>  
WHERE <Condition>
```

```
mysql> Select * from Student;  
+-----+-----+-----+-----+-----+  
| ROLL | NAME   | AGE | CLASS | CITY   |  
+-----+-----+-----+-----+-----+  
| 1001 | Pankaj | 6   | I     | Barabanki |  
| 1002 | Sunita | 7   | II    | Barabanki |  
| 1003 | Mukesh | 5   | I     | Barabanki |  
| 1004 | Ramesh | 8   | II    | Lucknow   |  
| 1005 | Yogesh | 9   | III   | Kanpur    |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

यहाँ * का अभिप्राय
समस्त columns से है |
और बिना condition
समस्त records
दिखायेगा |

```
mysql> Select * from Student Where city='Barabanki';  
+-----+-----+-----+-----+-----+  
| ROLL | NAME   | AGE | CLASS | CITY   |  
+-----+-----+-----+-----+-----+  
| 1001 | Pankaj | 6   | I     | Barabanki |  
| 1002 | Sunita | 7   | II    | Barabanki |  
| 1003 | Mukesh | 5   | I     | Barabanki |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

यहाँ condition है कि
जिनकी city 'Barabanki'
हो |

Table को access करना

Table से डाटा को access करने के लिए निम्न syntax होता है -

```
mysql> SELECT <Col Names> FROM <Table Name>  
WHERE <Condition>
```

```
mysql> Select Name, Class from Student where City (< 'Barabanki';  
+-----+-----+  
| Name   | Class |  
+-----+-----+  
| Ramesh | II    |  
| Yogesh | III   |  
+-----+-----+  
2 rows in set (0.00 sec)
```

यहाँ सिर्फ name और class ही दिखाना है लेकिन जिनकी city 'Barabanki' न हो |

```
mysql> Select Name, Roll, City, Age, Class  
-> from Student;  
+-----+-----+-----+-----+-----+  
| Name   | Roll  | City      | Age  | Class |  
+-----+-----+-----+-----+-----+  
| Pankaj | 1001  | Barabanki | 6    | I     |  
| Sunita | 1002  | Barabanki | 7    | II    |  
| Mukesh | 1003  | Barabanki | 5    | I     |  
| Ramesh | 1004  | Lucknow   | 8    | II    |  
| Yogesh | 1005  | Kanpur    | 9    | III   |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

यहाँ प्रदर्शन से पहले column के क्रम बदल दिए गए हैं |

Distinct keyword

```
mysql> select DISTINCT City from Student;
+-----+
| City          |
+-----+
| Barabanki     |
| Lucknow       |
| Kanpur        |
+-----+
3 rows in set (0.00 sec)
```

यहाँ DISTINCT keyword का प्रयोग करके city की Duplicacy हटा कर दर्शाया गया है | इससे यह पता चलता है की कल कितनी city प्रयोग हुई हैं |

Tables को देखना

```
mysql> show tables;
+-----+
| Tables_in_school |
+-----+
| student          |
+-----+
1 row in set (0.00 sec)
```

Database में उपस्थित समस्त tables की सूची प्रदर्शित कर देगा |

Pattern Matching

```
mysql>
mysql> Select * from Student Where
-> name like 'S%';
+-----+-----+-----+-----+
| ROLL | NAME   | AGE | CLASS | CITY      |
+-----+-----+-----+-----+
| 1002 | Sunita | 7   | II    | Barabanki |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

यहाँ like के साथ pattern में दो symbol प्रयुक्त होते हैं '%' और '_' जहाँ '%' एक साथ कई characters को check करता है जबकि '_' सिर्फ एक character को represent करेगा |

ऊपर के उदहारण में 'S' से शुरू होने वाले सरे नाम दिखाए |
जबकि नीचे वाले उदहारण में दूसरे अक्षर 'u' वाले सारे नाम दिखाए |

```
mysql> Select * from Student Where
-> name like '_u%';
+-----+-----+-----+-----+
| ROLL | NAME   | AGE | CLASS | CITY      |
+-----+-----+-----+-----+
| 1002 | Sunita | 7   | II    | Barabanki |
| 1003 | Mukesh | 5   | I     | Barabanki |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

इस उदहारण से आप '_' और '%' के मध्य अंतर आसानी से समझ सकते हैं |

यदि हमें second last अक्षर 's' चाहिए तो निम्न pattern लिखना होगा

Name like '%s_'

Table से Tableको बनाना

एक table से दूसरी table बनाने के लिए निम्न syntax है -

```
mysql>CREATE TABLE <TableName>  
        AS (SELECT <Cols> FROM <ExistingTable>  
        WHERE <Condition>);
```

```
mysql>  
mysql> create table bbkscl as (select name, age from student  
-> where city='Barabanki');  
Query OK, 3 rows affected (0.09 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select * from bbkscl;  
+-----+-----+  
| name   | age  |  
+-----+-----+  
| Pankaj | 6    |  
| Sunita | 7    |  
| Mukesh | 5    |  
+-----+-----+  
3 rows in set (0.00 sec)
```

एक table से दूसरी table बनाने के लिए दिए गए उदाहरण को ध्यान से देखिये।

अन्य SQL कमांड

- Select * from Student where city *in* ('Lucknow','Barabanki');
- Select * from Student where city *Not in* ('Lucknow','Barabanki');
- Select * from Student where age *between 5 and 7*;
- Select * from Student *Order by* name *DESC*;
- Select *5 * 6* from *DUAL* ;
- Select *avg(age)* from student; //ऐसे ही count, इत्यादि फंक्शन

बाकी के commands आप बुक को refer करके पढ़ सकते हैं |

जितना अभ्यास करेंगे उतना अधिक मज़बूत होंगे |

All the Best

धन्यवाद

और अधिक पाठ्य-सामग्री हेतु निम्न लिंक पर क्लिक करें -

www.pythontrends.wordpress.com

एक शुरुआत pythontrends

पाइथन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

लेख/Articles

छायाचित्र/Images

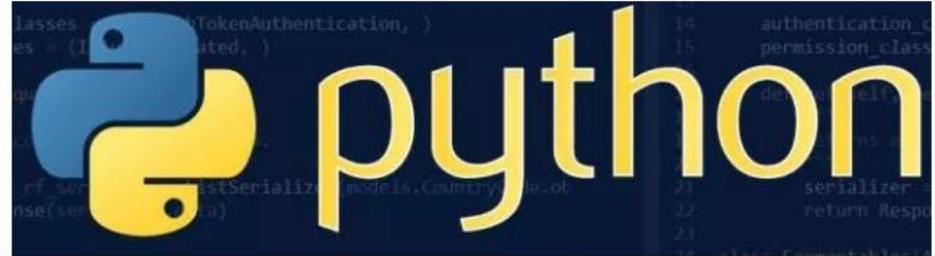
विडियो/Video

अध्यायवार पाठ्यसामग्री/Lesson wise
Study Material

उपयोगी लिंक्स / Useful Links

पाइथन प्रोग्राम/Python Programs

नमस्ते दोस्तों ! /Hello Friends!



यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रेजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेज़ी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेज़ी के कठिन शब्दों को

