Practical -2

MySQL Insert Records into Tables,and MySQL Constraints

- Display records/row from database
  Syntax : SELECT column name1, column name2… from tablename;

  Column name are the field names available in table
  Table name : Name of the table whose data needs to be display

  E.g 1 . select * from temp;
  It will show all the fields/attribute of table temp.

  Eg 2 select name from temp;
  It will show only the name attribute/field/column from table temp;

MySQL Constraint

MySQL CONSTRAINT is used to define rules to allow or restrict what values can be stored in columns. The purpose of inducing constraints is to enforce the integrity of a database.

MySQL CONSTRAINTS are used to limit the type of data that can be inserted into a table.

MySQL CONSTRAINTS can be classified into two types –

- column level : constraints can apply only to one column
- table level : constraints are applied to the entire table.

MySQL CONSTRAINT is declared at the time of creating a table.

MySQL CONSTRAINTs are

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

CONSTRAINT	DESCRIPTION
NOT NULL	In MySQL NOT NULL constraint allows to specify that a column can not contain any NULL value

UNIQUE

The UNIQUE constraint in MySQL does not allow to insert a duplicate value in a column. More than one UNIQUE column can be used in a table.

PRIMARY KEY

A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint creates **a unique index for accessing the table faster.**

FOREIGN KEY

A FOREIGN KEY in MySQL creates a link between two tables by one specific column of both tables. The specified column in one table must be a PRIMARY KEY and referred by the column of another table known as FOREIGN KEY.

CHECK

A CHECK constraint controls the values in the associated column. The CHECK constraint determines whether the value is valid or not from a logical expression.

DEFAULT

In a MySQL table, each column must contain a value (including a NULL). While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT

- NOT NULL Constraint & DEFAULT constraint

NOT NULLenforced that a column in a table is not allowed to store NULL values

DEFAULT is used to set a default value for a column and is applied using

Syntax :: DEFAULT defaultvalue

e.g
```
CREATE TABLE Persons (
 Name varchar(30) NOT NULL,
Age tinyint,
City varchar(20) DEFAULT ‘AHMD’);
```

2 insert into persons(age) values(23); → **error not null field name should not be null**

ERROR 1364 (HY000): Field 'name' doesn't have a default value

3 insert into persons(name,age) values(‘abcd’,23);

It takes the default value ‘AHMD’ in field city even it is not input

4 insert into persons(name,age) values ('aaaa',22),('bbbb',34);

Select * from persons;

```
+------+------+------+
| name | age  | city |
+------+------+------+
| abcd |   23 | AHMD |      -> default value ‘AHMD’ taken in field city.
| aaaa |   22 | AHMD |
| bbbb |   34 | AHMD |
+------+------+------+
```

- **Check Constraint**

  It is Column level constraint. Adding a CHECK CONSTRAINT on a column of a table, you can limit the range of values allowed to be stored in that column. Check constraint can also define at the end of table.

  E.g Age values must be grater than 18. Check constraint define at creation of table like

```
  mysql> create table per_tab1(name varchar(30) NOT NULL ,
      Age tinyint  CHECK(Age >=18) ,
      City varchar(20) DEFAULT  ‘AHMD’);
                OR
mysql>create table per_tab1(name varchar(30) NOT NULL ,
      Age tinyint
      City varchar(20) DEFAULT  ‘AHMD’,
     CHECK (Age>=18) );
```

  Checking constraint at the time of insertion of data , query for insert is written like

```
mysql> insert into per_tab1(name,age,city) values('Ronak sharma',10,'delhi');
   ERROR 3819 (HY000): Check constraint 'per_tab1_chk_1' is violated.
```

It shows constraint violate error as age is input 10 and 10 < 18.

```
mysql> insert into per_tab1(name,age,city) values('Ronak sharma',20,'delhi');
```

successfully inserted with check constraint

CHECK Constraint with LIKE Operator

Like operator is used, to set a format for storing values .
e.g in given book table check constraint with like operator set the format that book id start with B

```
mysql > create table book(bookid varchar(10) NOT NULL
CHECK(bookid LIKE 'B%'),
    bookname varchar(30),
```

);

- **Primary Key**
  The PRIMARY KEY constraint uniquely identifies each record in a table.
  primary key can consist of single or multiple columns (fields).
  Primary key is table level constraint. Unique key is column level constraint
  Table can have only one primary key , while table can have multiple unique key.

  e.g
  mysql> create table book3(bookid tinyint **primary key** , bookname varchar(30) , price int(4));

  bookid is become primary key which uniquely identified the record.

- **Foreign Key**
  A foreign key is a key used to link two tables together. **it** is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.

  e.g suppose table author contains information about author3.
  authid is primary key.
  Table book3 contains information about book. bookid is primary key in table.

  Syntax :

```
          FOREIGN KEY
 [index_name] (col_name, ...)
 REFERENCES tbl_name (col_name,...)
 [ON DELETE reference_option]
 [ON UPDATE reference_option]
```

  Col_name : name of column on which we create foreign key
  Tbl_name(col_name) : name of parent table where primary key is declared and col_name is field name of parent table.

  mysql>create table author3(authid tinyint ,authname varchar(30) , primary key(authid));

```
create table book3(bookid tinyint primary key , bookname
varchar(30) , price int(4) );
```

Two table author3 and book3 created.

New table called **auth_book** created which contains the relation between auth and book . i.e which author written which book.

Authid,bookid will become foreign key in auth_book table. And both will combinedly work as primary key.

```
mysql>
create table auth_book(authid tinyint,bookid tinyint,pub_date date
,
   -> foreign key(authid) references author3(authid),
   -> foreign key (bookid) references book3(bookid) ,
   -> primary key(authid,bookid) );
```

Values inserted into book3.

```
Mysql>insert into book3 values
(1,'dbms',400),(2,'network',500),(3,'datamining',600);
```

Values inserted into author3

```
insert into author3 values(1,'aaa'),(2,'bbb');
```

Values inserted into auth_book.

```
insert into auth_book values (1,1,'2002-05-18') , (1,2,'2005-09-23') ;
```

```
insert into auth_book values (4,1,'2002-05-18')  ;
```

- generate error as authid is foreignkey from table author3 on authid.
- Author3 table doesnot contain the value ‘4’ for the authid field. So it don’t allow to enter authid in auth_book the values which does not exists on parent table ‘s primary key.

- **Auto Increment Constraint**

MySQL allows you to set AUTO_INCREMENT to a column. Doing so will increase the value of that column by 1 automatically, each time a new record is added.

```
Mysql > create table temp(id  int AUTO_INCREMENT,
            name varchar(10),
            primary key(id) );
```

AUTO_INCREMENT column must be primary key.

```
Mysql> insert into temp(name)
values('abcd'),('rohan'),('sima'),('vihar');
```

Auto_increment column will automatically takes values.

```
Mysql> select *from temp;
```

EXCERCISE

Create the tables for the following:

Table Name: CLIENT_MASTER

Description: Used to store client information

Column Name	Data Type	Size	Default	Attributes
CLIENTNO	Varchar	6		Primary key/first letter must start with 'C'
NAME	Varchar	20		Not Null
CITY	Varchar	15		
STATE	Varchar	15		
BALDUE	Float	10,2		

able Name: PRODUCT_MASTER

Description: stores product information

Column Name	Data Type	Size	Default	Attributes
PRODUCTNO	Varchar	6		Primary key/ first letter must start with 'P'
DESCRIPTION	Varchar	15		NOT NULL
UNITMEASURE	Varchar	10		NOT NULL
QTYONHAND	Int	8		NOT NULL
REORDERVL	Int	8		NOT NULL
SELLPRICE	Float	8,2		cannot be 0
COSTPRICE	Float	8,2		cannot be 0

Table Name : SALESMAN_MASTER
Description: stores sales man information

Column Name	Data Type	Size	Default	Attributes

SALESMANNO	Varchar	6		Priamry key/ first letter must start with 'S'
SALESMANNAME	Varchar	30		Not Null
CITY	Varchar	20		
STATE	Varchar	20		
SALE_AMT	Float	8,2		cannot be 0

Table Name: SALES_ORDER
Description: store client order to the salesman

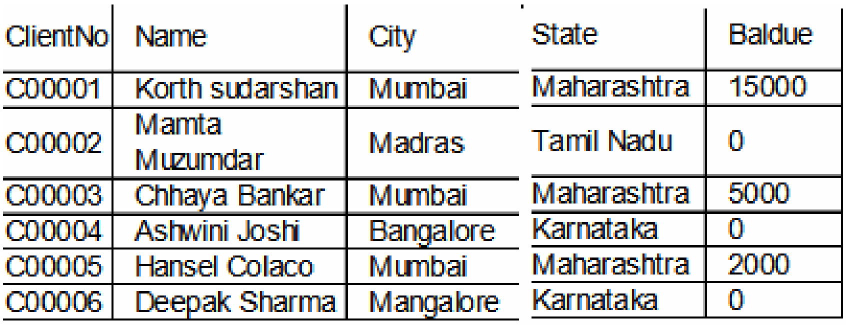
Column Name	Data Type	Size	Default	Attributes
ORDERNO	Varchar	6		Priamry key/ first letter must start with 'O'
CLIENTNO	Varchar	6		Foreign Key references ClientNo of Client_Master table
ORDERDATE	Date			Not Null
SALESMANNO	Varchar	6		Foreign Key reference SalesmanNo of Salesman_Master table
DELYTYPE	Enum	1		Delivery : part(P) / full(F)
DELYDATE	Date			
ORDERSTATUS	Enum			Values('In Process', 'Fulfilled','BackOrder','Cancelled')

Table Name: SALES_ORDER_DETAILS
Description: Used to store client's order details of each product ordered

Column Name	Data Type	Size	Default	Attributes
ORDERNO	Varchar	6		Foreign Key references OrderNo of Sales_Order table
PRODUCTNO	Varchar	6		Foreign Key references ProductNo of Product_Master table
QTYORDERED	Int	8		

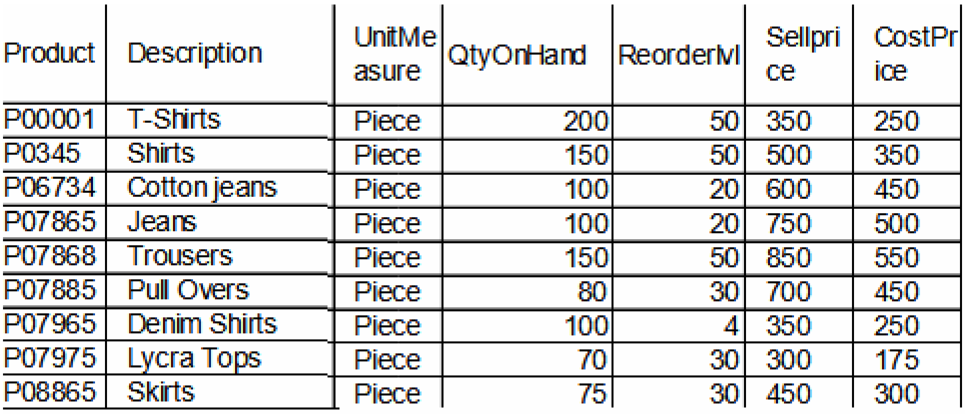
Insert Records into the table

Client_master



ClientNo	Name	City	State	Baldue
C00001	Korth sudarshan	Mumbai	Maharashtra	15000
C00002	Mamta Muzumdar	Madras	Tamil Nadu	0
C00003	Chhaya Bankar	Mumbai	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	Karnataka	0
C00005	Hansel Colaco	Mumbai	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	Karnataka	0

Product_Master



Product	Description	UnitMeasure	QtyOnHand	Reorderlvl	Sellprice	CostPrice
P00001	T-Shirts	Piece	200	50	350	250
P0345	Shirts	Piece	150	50	500	350
P06734	Cotton jeans	Piece	100	20	600	450
P07865	Jeans	Piece	100	20	750	500
P07868	Trousers	Piece	150	50	850	550
P07885	Pull Overs	Piece	80	30	700	450
P07965	Denim Shirts	Piece	100	4	350	250
P07975	Lycra Tops	Piece	70	30	300	175
P08865	Skirts	Piece	75	30	450	300

Salesman_Master

SalesmanNo	Name	City	State	Sale_amt
S00001	Aman	Mumbai	Maharashtra	3000
S00002	Omkar	Mumbai	Maharashtra	6700
S00003	Ray	Mumbai	Maharashtra	4000
S00004	Ashishh	Ahmedabad	Gujarat	3500

Sales_Order Table

OrderNo	ClientNo	OrderDate	SalesmanNO	DelyType	DelyDate	OrderStatus
O19001	C00001	2004-06-12	S00001	F	2004-07-20	In Process
O19002	C00002	2004-06-25	S00002	P	2004-06-27	Cancelled
O46865	C00003	2004-02-18	S00003	F	2004-02-20	Fulfilled

O19003	C00001	2004-04-03	S00001	F	2004-04-07	Fulfilled
O46866	C00004	2004-05-20	S00002	P	2004-05-22	Cancelled
O19008	C00005	2004-05-24	S00004	F	2004-07-26	In Process

Sales_Order_Details Table

OrderNo	ProductNo	Qtyorder
O19001	P00001	4
O19001	P07965	2
O19001	P07885	2
O19002	P00001	10
O46865	P07868	3
O46865	P07885	3
O46865	P00001	10
O46865	P0345	4
O19003	P0345	2
O19003	P06734	1
O46866	P07965	1
O46866	P07975	1
O19008	P00001	8
O19008	P07975	5