Practical -3

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 NULL enforced 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 IN operator

MySQL CHECK CONSTRAINT can be applied to a column of a table, to set a limit for storing values within a range, along with IN operator.

e.g

```
Mysql>  create table per_tab2 (name varchar(30) NOT NULL,
 Age tinyint CHECK(Age>=18),
city varchar(20) DEFAULT 'AHMD' CHECK (city in
      ('AHMD','delhi','mumbai','pune'))
 );
```

It will create table with name as NOT NULL constraint,
Check constraint on Age field check Age must be >=18 while insert/update record
Check constraint in city filed contains IN opration that is value of city can be either 'AHMD', or delhi or mumbi or pune. Any other values apart from it is not allowed

Insert value

```
Mysql>  insert into per_tab2(name,Age,city) values('kirti patel',29,'pune');
```

CHECK Constraint with LIKE Operator , BETWEEN 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
2. Between operator with check constraint set the range of value that is inserted for price field that is between 100 to 5000

```
mysql > create table book(bookid varchar(10) NOT NULL
CHECK(bookid LIKE 'B%'),
    bookname varchar(30),
    price int(4) CHECK (price between 100 AND 5000 )
   );
```

CHECK Constraint with AND OR operator

It apply the and or condition on the values, for multiple field values

e.g

```
mysql> create table author (aid tinyint, name varchar(30),
   country varchar(10),
    city varchar(15),
   CHECK ((country='india' and city='mumbai') OR (country='india'
AND city='delhi') )
   );
```

- **Unique Constraint**

  The UNIQUE constraint creates an index such that, all values in the index column must be unique. It stores the unique values for field.

  e.g

  create table book1 (bookid tinyint , bookname varchar(30),price int(4) , unique(bookname));

  name of book must be unique, same values can not be inserted second time.

  mysql> insert into book1 values(2,'dbms',400);
  ERROR 1062 (23000): Duplicate entry 'dbms' for key 'book1.bookname'

- **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 author. 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.

  When an UPDATE or DELETE operation affects a key value in the parent table that has matching rows in the child table, the result depends on the referential action specified by ON UPDATE and ON DELETE

  CASCADE: Delete or update the row from the parent table and automatically delete or update the matching rows in the child 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		
PINCODE	Int	8		
STATE	Varchar	15		
BALDUE	Decimal	10,2		

Table 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
PROFITPERCENT	Decimal	4,2		NOT NULL
UNITMEASURE	Varchar	10		NOT NULL
QTYONHAND	Int	8		NOT NULL
REORDERVL	Int	8		NOT NULL
SELLPRICE	Decimal	8,2		NOT NULL, cannot be 0
COSTPRICE	Decimal	8,2		NOT NULL, 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	20		Not Null
ADDRESS	Varchar	30		Not Null
CITY	Varchar	20		
PINCODE	Int	8		
STATE	Varchar	20		
SALE_AMT	Decimal	8,2		Not Null, cannot be 0
TGTTOGET	Decimal	6,2		Not Null, cannot be 0
YTDSALES	Decimal	6,2		Not Null
REMARKS	Varchar	60		

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
DELYADDR	Varchar	25		
SALESMANNO	Varchar	6		Foreign Key reference SalesmanNo of Salesman_Master table
DELYTYPE	Char	1		Delivery : part(P) / full(F)
DELYDATE	Date			
ORDERSTATUS	Varchar	10		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		
QTYDISP	Int	8		
PRODUCTRATE	Decimal	10,2		

Insert Records into the table

Client_master

ClientNo	Name	City	Pincode	State	Baldue
C00001	Korth sudarshan	Mumbai	400054	Maharashtra	15000
C00002	Mamta Muzumdar	Madras	780001	Tamil Nadu	0
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
C00005	Hansel Colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	560050	Karnataka	0

PRODUCT_MASTER

Product	Description	Profitpercent	UnitMeasure	QtyOnHand	Reorderlvl	Sellprice	CostPrice
P00001	T-Shirts	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	500	350
P06734	Cotton jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	4	350	250
P07975	Lycra Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

SALESMAN_MASTER

Salesman No	Name	Address2	City	Pincode	State	Sale_amt	Tgttoget	YtdSales
S00001	Aman	A/14 Worli	Mumbai	400002	Maharashtra	3000	1000	50

S00002	Omkar	C-65 Narima	Mumbai	400001	Maharash tra	3000	2000	100
S00003	Ray	P-7 Bandra	Mumbai	400032	Maharash tra	3000	2000	100
S00004	Ashishh	A/5 Juhu	Mumbai	400044	Maharash tra	3500	2000	150

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	Qtydisp	Productrate
O19001	P00001	4	4	250
O19001	P07965	2	1	250
O19001	P07885	2	1	450
O19002	P00001	10	5	250
O46865	P07868	3	3	550
O46865	P07885	3	3	450
O46865	P00001	10	10	250
O46865	P0345	4	4	350
O19003	P0345	2	2	350
O19003	P06734	1	1	450
O46866	P07965	1	0	250
O46866	P07975	1	0	175
O19008	P00001	8	6	250
O19008	P07975	5	4	175