iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VI **Month of publication:** June 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Review of Integrity Constraint in MySQL

Deepika[#1], Mr. Anil Arora[#2]
[1]M. Tech. Scholar, [2]Assistant Professor, Department of Computer Science & Engineering
Gateway Institute of Engineering & Technology (GIET), Sonepat

Abstract*— For the semantic correctness of database records, Integrity constraints or data constraints are proper representations of invariant conditions. Constraints can be articulated in declarative languages such as data log, predicate logic, or SQL. All business applications run with predefined rules and these rules are also valid to business data and they should not be violated. MySQL has a special feature call data constraint or integrity constraint that was useful at the time of creation of data structure. Business rules, which are imposed on data being stored in a table, are called constraints. for permanent storage ,Constraints super control the data being entered into a table. In this paper we have to learning the two types of data constraints that can be applied to data being inserted into a Mysql table. 1). I/O constraint and 2). Business constraint.
Keywords*— Data Constraint, I/O constraint, Business Constraint.

I. INTRODUCTION

MySQL allow data constraints to be attached to table column level via sql syntax that check data for integrity earlier stage. Once data constraints are component of a table column construct, the data being entered into a table column against the data constraints checked by mysql database engine. Data stored in the table column , if the data surpass this check, else the data are rejected. Even if a single column of the record being entered into table not satisfy the constraint and the entire record is not stored in the table and entire record is discarded.

Constraints can be clear either at the table level or column level. If the data constraint is attached to a specific cell in a table reference to the contents of another cell in the table, then user will have to use table level constraints. In column level constraints, if the constraints are defined all along with the column definition then these are called column level constraints. Table level constraints are stored as a branch of the global table definition.

II. TYPES OF CONSTRAINTS

The main problem of integrity for database design and control, as attested by many early publications (e.g., Bernstein & Blaustein, 1982; Bernstein, Blaustein, & Clarke, 1980; Codd, 1970, 1979; Eswaran & Chamberlin, 1975; Fraser, 1969; Hammer & McLeod, 1975; Hammer & Sarin, 1978; Nicolas, 1978, 1982; Wilkes, 1972); later ones are too frequent to mention. Database semantics are expressed as invariant properties continuing across updates had first been proposed by Minsky (1974). According to Florentin (1974), integrity constraints expressed as predicate logic statements. According to Stonebraker (1975), formulating and checking integrity constraints declaratively intended as SQL-like queries.

To direct database design, Functional dependencies (Armstrong, 1974; Codd, 1970) are a essential type of constraints. Referential integrity has been component of the 1989 SQL ANSI and ISO standards (McJones, 1997). To state integrity constraints declaratively (Date & Darwen, 1997), the CHECK and ASSERTION constructs (i.e., table-bound and table-independent SQL query conditions) introduced as the most general means by the SQL2 standard (1992). Since the 1990s, in commercial databases, uniqueness constraints, foreign keys, and complex queries involving EXISTS and NOT became common features. Thus, In most relational databases, randomly general and complex integrity constraints can now be expressed and evaluated. However, For the most part of them present professional support only for the three simple kind of declarative constraints:

A. Domain Constraints

In table columns, restrictions on the acceptable range of attribute values of tuples , including scalar SQL data types and subsets as well as choices for default and null values.

B. Uniqueness Constraints

As imposed by the UNIQUE construct on single columns, and UNIQUE INDEX and PRIMARY KEY on any mixture of one or numerous columns in a table, checking multiple occurrences of values or combinations.

C. Foreign Key Constraints

Identical column values need for establishing a connection between the tuples of two tables,. For example, a foreign key on column emp of relation works_in needs that the emp value of each tuple of works_in must take place in the emp_id column of table employee, and that the referenced column (emp_id in the example) has been declared as primary key.

For additional common constraints, SQL manuals usually suggested using procedural triggers or stored procedures as an alternative of declarative constructs. This is because such constraints may occupy nested quantifications over huge extents of several tables.

Thus, their cost of evaluation can easily become prohibitively high. However, declaratively does not need to be sacrificed for efficiency, as shown by many methods of simplified integrity checking as cited in this survey. They are all based on the seminal paper (Nicolas, 1982).

III. TYPES OF CONSTRAINTS

In this study we have determined two types of constraints linked to MySQL. one is called an I/O constraint (Input/output) and the other one is called Business constraint.

A. I/O Constraints

I/O constraints are those constraints which establish the speed at which data can be inserted or extracted. The three distinctly constraints of I/O data constraints :

1) *The Primary Key Constraint:* A primary key constraints are those in which one or more column(s) in a table used to individually identify each row in table. Primary key values must not be null and must be unique across the column(s). A single column primary key is called a simple key. A multi-column primary key is called a composite primary key. In short Primary key = Unique + Not Null.

Ex1:

```
Create table student
(roll_no number(5) primary key,
name varchar2(50));
```

Ex2:

```
Create table stud_result
(rollno number(5),
name varchar2(50),
paper_no number(3),
marks number(3),
primary key(rollno, paper_no));
```

a) Features of primary key:

i) The value of primary key column is not null (compulsory). and unique (no duplication).

ii) Primary key facilitate to find one record from another record and facilitates in relating table with one another.

iii) It creates unique index automatically.

iv) In a composite primary key , one table can join up to 16 columns.

2) *The Foreign Key Constraint:* Foreign key correspond to relationships between tables. A foreign key is a column (or a group of columns) whose values are obtained from the primary key or unique key of some other table. The foreign key is also identified as referential integrity constraints. Foreign table or Detail table are those table in which the foreign key is described. Primary table or Master table are those table that defines the primary or unique key and is mentioned by foreign key.

Ex1:

```
Create table stud_result
```

```
(roll_no number(5) references student(roll_no),
paper_no number(3),
marks number(3),
primary key(roll_no,paper_no));
```

Ex2:

```
Create table stud_result
(rollno number(5),
name varchar2(50),
paper_no number(3),
marks number(3),  primary key (rollno, paper_no) ,
foreign key(roll_no) references student);
```

a) Features of foreign key

i) The detail table and data type of the appropriate column in the master must be same.

ii) Parent that being mentioned has to be unique or primary key.

iii) Child may have duplicates and null but unless it is specified.

iv) Foreign key constraint can be specified on child but not on parent.

v) If corresponding records are available in child table then deleting record from the parent table is not allowed.

vi) If child records exist then master table cannot be updated.

3) *The Unique Key Constraint:* The unique key is similar to a primary key, except that the reason of a unique key is to guarantee that the information in the column for each record is unique. Unique key constraint allow null values in the unique key column.

Ex1:

```
Create table course
(course_id number(3) primary key,
name varchar2(30) unique);
```

Ex2:

```
Create table course
(course_id number(3) primary key,
name varchar2(30),unique(name));
```

a) Features of unique key:

i) Duplicate values are not allowed, but can allow null values.

ii) It automatically creates unique index.

iii) A table can contain more than one unique key which is not possible in the primary key.

iv) One table can join up to 16 columns in a composite unique key.

B. Business Constraint

The rules are necessary to run the business applications, while storing business data into the oracle database, data should track the business rules. In oracle business rules can be implemented using check constraints.

When any table write operation is carried out business rule validation checks are performed. Any insert or update statement causes the significant check constraint to be evaluated. The check constraint must be fulfilled for the write operation to be succeed. Thus in tables check constraints make sure the integrity of the data.

1) *NOT NULL Constraint*: InMySQL, NOT NULL column level constraints, in which such table column cannot be left empty. Column becomes a mandatory column when a column is stated as Not Null than. That means null value is not allowed for that particular column. Not Null constraints can be defined only at the column level.

Ex:

```
Create table student
(roll_no number(5) primary key,
name varchar2(50) not null ));
```

2) *Check Constraint*: In MySQL, Business rules can be distinct using check constraint. Firstly it will check constraint before it insert or update column data values. If the data value according to the constraint rule then data will be effectively inserted or updated. In any case,data will not be successfully inserted or updated and statement will be rolled back, if constraint rules are violated data.

Check constraint must be precise as a logical expression that evaluates to be true or false.

Ex1:

```
Create table stud_result
(roll_no number(5),
paper_no number(3),
marks number(3) check(marks<=100));
```

Ex2:

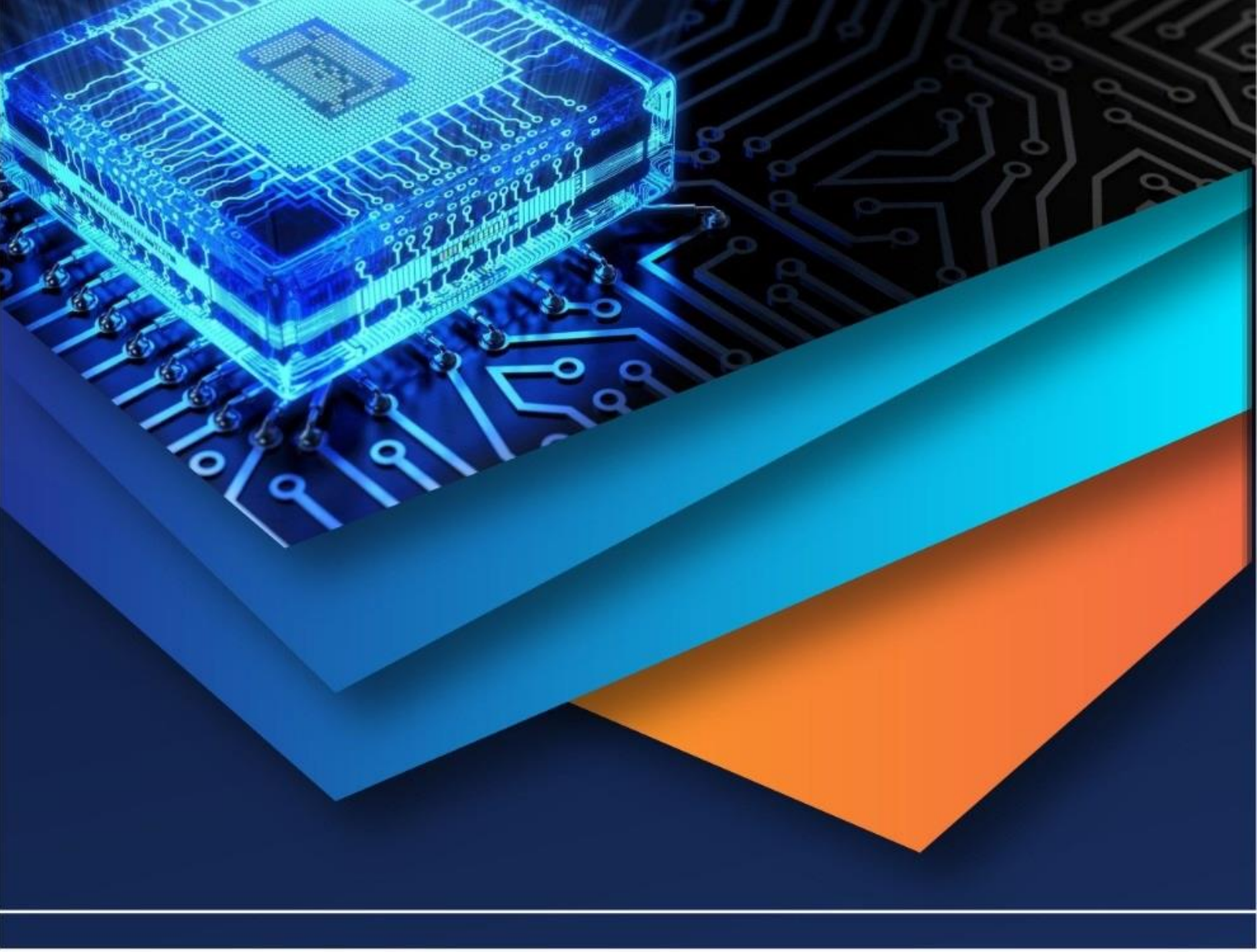
```
Create table stud_result
(roll_no number(5),
paper_no number(3),
total_marks number(3),
obt_marks number(3),
check(obt_marks<=total_marks));
```

IV. CONCLUSIONS

From the above we can conclude that MySQL Data Constraints are wide conceptualized. This research finds out that first all business data being gathered stored and analysed then all business of the world runs on business data. Business managers conclude to ensure its integrity, a set of business rules that must be applied to their data prior to it which is in stored in the database/table. Only that data should be stored for future analysis which satisfies the conditions (rules) set. It must be rejected if the data gathered fails that means it is not satisfy the conditions (rules) set. This states that the data stored in a table will be valid and have integrity. Thus, we can say that a data constraint provides integrity and data level security.

REFERENCES

[1] E. F. Codd, "A Relational Model of Data for Large Common Data Banks ", Comm. of the ACM (1970).
[2] Dr. P. M. Dolia, Mr. K. U. Gundigara, Ms. V. H. Mehta, Database Concepts SQL & PL/SQL Programming. Ahmedabad: Nirav Prakashan.
[3] Dr. Arvind Mohan Parashar, Chandresh Shah, Saurab Mishra, Sunilkumar Ojha Computer Science & Application. Agra: Upkar Prakashan.
[4] Ivan Bayross (2009). The programming language of Oracle. New Delhi: BPB publications
[5] Jeffrey R. Shapiro.SQL Server 2000:The Complete Reference- McGraw-Hill
[6] J. Widom and S. Ceri. Active Database Systems: Triggers and Rules for Higher Database Processing. Morgan Kaufmann, San Francisco, California, 1996.
[7] J.D. Ullman and J. Widom. A First Course in Database Systems. Prentice Hall, Upper Saddle River, New Jersey, 1997.
[8] Webopedia: 2005, Data integrity http://www.pcwebopaedia.com/TERM/d/data_integrity.html
[9] Wikipedia: 2005, PL/SQL http://en.wikipedia.org/wiki/PL/SQL



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY