International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 6, June-2015, pp: (118-122), Impact Factor: 1.252, Available online at: www.erpublications.com

Review of Integrity Constraint Available in Oracle

Ritu Raj[1], Kirti Bhatia[2]
[1,2]M. Tech. Scholar, Department of CSE, Sat Kabir Institute of Technology & Management (SKITM), Bahadurgarh, Haryana, India

Abstract: **Integrity constraints (or simply "constraints") are formal representations of invariant conditions for the semantic correctness of database records. Constraints can be expressed in declarative languages such as data log, predicate logic, or SQL. All business applications run with predefined rules, these rules are also applicable to business data and they should not be violated. Oracle provides a special feature call data constraint or integrity constraint that was applied at the time of creation of data structure. Business rules, which are enforced on data being stored in a table, are called constraints. Constraints super control the data being entered into a table for permanent storage. In this paper we have to study the two types of data constraints that can be applied to data being inserted into an oracle table. 1). I/O constraint and 2). Business constraint.**

Keywords: Data Constraint, I/O constraint, Business Constraint.

I. INTRODUCTION

Oracle permits data constraints to be attached to table column level via sql syntax that check data for integrity earlier stage. once data constraints are part of a table column construct, the oracle database engine checks the data being entered into a table column against the data constraints. if the data pass this check, it stored in the table column, else the data are rejected. even if a single column of the record being entered into table fails a constraint and the entire record is rejected and not stored in the table. Constraints can be defined either at the column level or table level. if the constraints are defined along with the column definition, they are called column level constraints. 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. Table level constraints are stored as a part of the global table definition.

II. BACKGROUND OF CONSTRAINTS

Integrity has always been an important issue 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 numerous to mention. Expressing database semantics as invariant properties persisting across updates had first been proposed by Minsky (1974). Florentin (1974) suggested expressing integrity constraints as predicate logic statements. Stonebraker (1975) proposed formulating and checking integrity constraints declaratively as SQL-like queries.

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

• **Domain Constraints:** Restrictions on the permissible range of attribute values of tuples in table columns, including scalar SQL data types and subsets thereof, as well as options for default and null values.

• **Uniqueness Constraints:** As enforced by the UNIQUE construct on single columns, and UNIQUE INDEX and PRIMARY KEY on any combination of one or several columns in a table, preventing multiple occurrences of values or combinations thereof.

• **Foreign Key Constraints:** For establishing a relationship between the tuples of two tables, requiring identical column values. For instance, a foreign key on column emp of relation works_in requires that the emp value of each tuple of works_in must occur 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 more general constraints, SQL manuals usually recommend using procedural triggers or stored procedures instead of declarative constructs. This is because such constraints may involve nested quantifications over huge extents of several tables. Thus, their evaluation can easily become prohibitively costly. 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 considered two types of constraints related to oracle. The first one is called an I/O constraint (Input/output) and the second one is called Business constraint.

[A]. I/O Constraints:

Constraints which determine the speed at which data can be inserted or extracted are called I/O constraints. The I/O data constraints are further divided into three distinctly constraints.

1) The Primary Key Constraint:

A primary key is one or more column(s) in a table used to uniquely 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));
```

Features of primary key:

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

■ Primary key help to find one record from another record and helps in relating table with one another.

■ It automatically creates unique index.

■ One table can combine up to 16 columns in a composite primary key.

2) The Foreign Key Constraint:

Foreign key represents relationships between tables. A foreign key is a column (or a group of columns) whose values are derived from the primary key or unique key of some other table. The foreign key is also known as referential integrity constraints.

The table in which the foreign key is defined is called foreign table or Detail table. The table that defines the primary or unique key and is referenced by foreign key is called primary table or master table.

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);
```

Features of foreign key:

■ The data type of the relevant column in the master and detail table must be same.

■ Parent that being referenced has to be unique or primary key.

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

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

■ Deleting record from the parent table is not allowed if corresponding records are available in child table.

■ Master table cannot be updated if child records exist.

3) The Unique Key Constraint:

The unique key is similar to a primary key, except that the purpose of a unique key is to ensure that the information in the column for each record is unique. Unique key constraint allows 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));
```

Features of unique key:

■ It does not allow duplicate values, but can allow null values.

■ It automatically creates unique index.

■ A table can have more than one unique key which is not possible in the primary key.

■ One table can combine up to 16 columns in a composite unique key.

[B]. Business Constraint:

All business applications are running based on rules, while storing business data into the oracle database data should follow the business rules. Business rules can be implemented in oracle by using check constraints.

Business rule validation checks are performed when any table write operation is carried out. Any insert or update statement causes the relevant check constraint to be evaluated. The check constraint must be satisfied for the write operation to succeed. Thus check constraints ensure the integrity of the data in tables.

1) NOT NULL Constraint:

Oracle provides NOT NULL column level constraints, which ensures that such table column cannot be left empty. When a column is defined as Not Null than that column becomes a mandatory column. It implies that the value of the particular column cannot be null. 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:

Business rules can be defined in oracle by using check constraint. Before inserting or updating column data values, first it will check constraint, if the data value satisfies the constraint rule then data will be successfully inserted or updated. In any case if constraint rules are violated data will not be successfully inserted or updated and statement will be rolled back.

Check constraint must be specified as a logical expression that evaluates to 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 **Oracle Data Constraints** are wide conceptualized. This research finds out that all business of the world runs on business data being gathered stored and analysed. Business managers determine a set of business rules that must be applied to their data prior to it being stored in the database/table to ensure its integrity. Only data, which satisfies the conditions (rules) set, should be stored for future analysis. If the data gathered fails to satisfy the conditions (rules) set, it must be rejected. This ensures that the data stored in a table will be valid and have integrity. Thus, we can say that data constraints are provides integrity and data level security.

REFERENCES

Books:

[1]. Jose' A Ramalho (2007). Oracle8i .New Delhi: BPB publications.
[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.

Web sites:

1. Data Integrity [On-Line] Available at: http://docs.oracle.com/cd/E11882_01/server.112/e25789/datainte.htm

2. Oracle Constraints types [On-Line] Available at:
http://www.dba-oracle.com/concepts/constraints_foreign_key_ unique_not_null.htm