**INTEGRITY CONSTRAINTS**

* It is used to impose business rules to Dbs.
* It allows entering only valid data.

Oracle uses integrity constraints to prevent invalid data entry into the base tables of the database. You can define integrity constraints to enforce the business rules that are associated with the information in a database. If any of the results of a DML statement execution violate an integrity constraint, Oracle rolls back the statement and returns an error.

For example, assume that you define an integrity constraint for the SAL column of the EMP table. This integrity constraint enforces the rule that no row in this table can contain a numeric value greater than 10,000 in this column. If an INSERT or UPDATE statement attempts to violate this integrity constraint, Oracle rolls back the statement and returns an informative error.

**Advantages of Integrity Constraints**

Integrity constraints are not the only way to enforce data integrity rules on the data of your database. You can also

* enforce business rules in the code of a database application
* use stored procedures to completely control access to data
* enforce business rules using triggered stored database procedures

## **Types of Integrity Constraints**

The integrity constraints that you can use to impose restrictions on the input of column values can be of the following types:

* NOT NULL constraints
* UNIQUE key constraints
* PRIMARY KEY constraints
* FOREIGN KEY (referential) constraints
* CHECK constraints

**Level of constraint**: -

**1)** **Column level**: -

🡪 Used to define constraints next in column name

🡪 Define with each column.

🡪 Composite key cannot be defined.

2) **Table Level**: -

🡪Defining constraints after defining all columns.

🡪Not Null cannot be defined.

The Constraint clause can appear in

* CREATE Table
* ALTER Table

**Note:** Oracle does not support Constraint on Column or attributes whose types USER\_DEFINED OBJECT, NESTED TABLE, VARRAY, REF, LOB

**Exceptions**:

• **NOT NULL** Constraint are supported for a column or attributes whose type is USER\_DEFINED object VARRAY, REF, LOB.

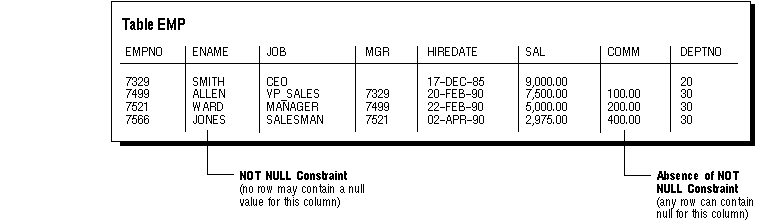
**NOT NULL Constraint**:

By default, all columns in a table allow nulls (the absence of a value). A NOT NULL constraint requires that no nulls be allowed in a column of a table.

* Used to suppress null values into columns.
* Data must be entered.
* Duplication values allowed.
* NOT NULL should be defined only at COLUMN Level.

**Example**:

For example, you can define a NOT NULL constraint to require that a value be input in the ENAME column for every row of the EMP table.



**Syntax to define a Not Null constraint:**

[CONSTRAINT constraint name] NOT NULL

Ex: CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINT EMP\_EMPNO\_NN NOT NULL,

ENAME VARCHAR2(10) CONSTRAINT EMP\_ENAME\_NN NOT NULL,

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2));

**UNIQUE Constraints**:

* Used to suppress duplicate values into columns.
* Accepts NULL values.
* A table can have more than one UINQUE key which is not possible in Primary Key.
* Unique key can define on more than one column (i.e composite unique key).
* A composite key UNIQUE key is always defined at the table level only.
* One table can have more than one Composite UNIQUE Key.
* Oracle creates an indexed automatically.

**Restrictions**:

* Unique key cannot be implemented on columns having....

**.LOB .LONG .LONG RAW .VARRAY .NESTED TABLE**

**.OBJECT .BFILE .REF .TIMESTAMP.**

* A composite UNIQUE key cannot have more than 32 columns.

**UNIQUE KEY Constrain Defined at Column Level**

**Syntax**:

<Column Name> <Data Type>(<Size>) UNIQUE

**Example**:

CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINTS EMP\_EMPNO\_U UNIQUE,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2));

Ex2:

CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINTS EMP\_EMPNO\_U UNIQUE CONSTRAINTS EMP\_EMPNO\_NN NOT NULL,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2));

**UINQUE Constraint Defined At the Table Level**

**Syntax**:

CREATE TABLE tableName

(<ColumnName1><Datatype>(<Size>),

<ColumnName2> <Datatype>(<Size>),

UNIQUE(<ColumnName1>,[ColumnName2,….]));

Ex: CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINTS EMP\_EMPNO\_NN NOT NULL,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MOBILE\_NO NUMBER(10),

EMAIL\_ID VARCHAR2(20),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2),

CONSTRAINTS EMP\_EMPNO\_U UNIQUE(EMPNO),

CONSTRAINTS EMP\_MOBILE\_NO\_U UNIQUE(MOBILE\_NO),

CONSTRAINTS EMP\_EMAIL\_ID\_U UNIQUE(EMAIL\_ID));

**Composite Unique Key:**

CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINT EMP\_EMPNO\_NN NOT NULL,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MOBILE\_NO NUMBER(10),

EMAIL\_ID VARCHAR2(20),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2),

CONSTRAINT EMP\_EMPNO\_U UNIQUE(EMPNO),

CONSTRAINT EMP\_MOBILE\_NO\_EMAIL\_ID\_U UNIQUE(MOBILE\_NO,EMAIL\_ID)

);

**PRIMARY KEY constraint**:

• Used to define key column of table.

• It will not accept Null Values and Duplicate values,

• It is provided with an automatic index.

• A Primary Key Constraint combines a NOT NULL and Unique behavior in one declaration.

**Restrictions**:

* Only one Primary key Or Composite Primary Key is allowing per table.
* PRIMARY KEY cannot be implemented on columns having….

**. LOB. LONG. LONG RAW. VARRAY. NESTED TABLE. OBJECT. BFILE. REF. TIMESTAMP.**

* A composite PRIMARY KEY cannot have more than 32 columns
* PRIMARY KEY cannot support in Nested Object.

**PRIMARY KEY Constraint Defined At Column Level**:

<Column Name> <Data Type>(<Size>) PRIMARY KEY

Example:

CREATE TABLE EMP

(EMPNO NUMBER(4) CONSTRAINTS EMP\_EMPNO\_PK PRIMARY KEY,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MOBILE\_NO NUMBER(10),

EMAIL\_ID VARCHAR2(20),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2));

**PRIMARY KEY Constraint Defined At the Table Level**:

**Syntax**.

CREATE TABLE tableName

(<ColumnName1><Datatype>(<Size>),

<ColumnName2><Datatype>(<Size>),

PRIMARY KEY(<ColumnName1>,[ColumnName2,….]));

**Example**:

CREATE TABLE EMP

(EMPNO NUMBER(4),

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MOBILE\_NO NUMBER(10),

EMAIL\_ID VARCHAR2(20),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2),

CONSTRAINTS EMP\_EMPNO\_PK PRIMARY KEY(EMPNO));

**Composite Primary Key:**

Ex:

CREATE TABLE EMP

(EMPNO NUMBER(4),

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MOBILE\_NO NUMBER(10),

EMAIL\_ID VARCHAR2(20),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(2),

CONSTRAINTS EMP\_EMPNO\_MOBILE\_EMAIL\_PK PRIMARY KEY(EMPNO,MOBILE\_NO,EMAIL\_ID));

**FOREIGN KEY (Or) REFERENTAIL INTEGRITY Constraint**:

* Foreign Key represent relationships between tables.
* A Foreign Key is a column or group of columns whose values are derived from the Primary Key or Unique Key.
* Foreign Key is column(s) that references a column(s) of a table and it can be the same table also.
* The Table or View constraining the Foreign key is called the Child object.
* Child may have duplicates and nulls but unless it is specified.
* A Foreign Key constraint can be defined on a single key column either column level (In line) or Table level (Out of line) column.
* A composite Foreign Key on attributes should be declared at table level or out of line.
* A Composite Foreign Key constraint, must refer to a composite unique key or a Composite Primary Key in the Parent Table.

**Restriction**:

• Master table cannot be update if child record exists.

• The Foreign Key column cannot be applied on….

**.LOB .LONG .LONG RAW .VARRAY .NESTED TABLE .OBJECT .BFILE .REF .TIMESTAMP.**

• A composite Foreign Key cannot have more than 32 columns.

• A Child and parent table must be on same database.

• To enable Referential Integrity across nodes of a distributed database triggers are used.

**Note**:

l)FOREIGN KEY identifies the column or combination of columns in the child table that makes up of the Foreign key.

2)REFERENCES identifies the parent table and the column or combination of columns that make up the referenced key.

**ON DELETE Clause**:

• CASCADE option used to remove the child table record automatically, when parent record is removed.

• Specify SET NULL if we want Oracle to convert dependent FOREIGN KEY values to NULL.

**Steps to be Followed for Creating References Constraint**:

Step1:

CREATE TABLE DEPT

(DEPTNO NUMBER (2)

CONSTRAINTS DEPT\_DEPTNO\_PK PRIMARY KEY,

DNAME VARCHAR2(14),

LOC VARCHAR2(13));

Step2: **Create Detail/Child/Sub/Dependent Table**

• These are Tables which can contain Primary Key of their own as well as Foreign key's referring to other Primary Master's or to themselves.

**Ex:**

CREATE TABLE EMP

(EMPNO NUMBER (4)

CONSTRAINT EMP\_EMPNO\_PK PRIMARY KEY,

ENAME VARCHAR2(10)

CONSTRAINT EMP\_ENAME\_NN NOT NULL,

JOB VARCHAR2(9),

MGR NUMBER (4)

CONSTRAINT EMP\_MGR\_FK REFERENCES EMP(EMPNO),

HIREDATE DATE,

SAL NUMBER (7,2),

COMM NUMBER (7,2),

DEPTNO NUMBER (2)

CONSTRAINT EMP\_DEPTNO\_FK REFERENCES DEPT(DEPTNO));

**EX2: Table level**

CREATE TABLE EMP

(EMPNO NUMBER (4)

CONSTRAINT EMP\_EMPNO\_PK PRIMARY KEY,

ENAME VARCHAR2(10)

CONSTRAINT EMP\_ENAME\_NN NOT NULL,

JOB VARCHAR2(9),

MGR NUMBER (4),

HIREDATE DATE,

SAL NUMBER (7,2),

COMM NUMBER (7,2),

DEPTNO NUMBER (2),

CONSTRAINT EMP\_MGR\_FK

FOREIGN KEY(MGR) REFERENCES EMP(EMPNO),

CONSTRAINT EMP\_DEPTNO\_FK FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO));

EX3:

CREATE TABLE EMP

(EMPNO NUMBER (4)

CONSTRAINT EMP\_EMPNO\_PK PRIMARY KEY,

ENAME VARCHAR2(10)

CONSTRAINT EMP\_ENAME\_NN NOT NULL,

JOB VARCHAR2(9),

MGR NUMBER (4),

HIREDATE DATE,

SAL NUMBER (7,2),

COMM NUMBER (7,2),

DEPTNO NUMBER (2),

CONSTRAINT EMP\_MGR\_FK

FOREIGN KEY(MGR) REFERENCES EMP(EMPNO) ON DELETE SET NULL,

CONSTRAINT EMP\_DEPTNO\_FK FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO) ON DELETE CASCADE);

**CHECK Constraint**:

* Used to impose a conditional rule on a table column.
* It defines condition that each row must satisfy.
* A single column can have multiple CHECK constants that can reference the column in the definitions.
* There is no limit to the number of CHECK constraints that can be defined on a column.
* The CHECK constrains can be defined at the column level or Table level.

**Restrictions**:

* The constructs that cannot be include are
* Queries to refer to values in other rows
* References to the CURRVAL, NESTVAL, LEVEL or ROWNUM.
* Calls to functions SYSDATE, UID, USER, USERENV.
* Date constant that are not fully specified.

**Example**:

CREATE TABLE Dept

(Deptno number(2) constraint dno\_pk PRIMARY KEY

constraint Deptno\_Chk

CHECK(Deptno BETWEEN 10 and 99),

Dname varchar2(15) constraint dname\_nn NOT NULL

constraint Dname\_Chk CHECK(Dname=UPPER(Dname)),

Loc varchar2(15) default 'NEW YORK'

constraint Loc\_Chk CHECK

(Loc IN('NEW YORK','DALLAS','BOSTON','CHICAGO'))

);

CREATE TABLE Emp

( Empno Number(4) Constraint Empno\_Pk PRIMARY KEY,

Ename Varchar2(20) Constraint ename\_NN NOT NULL

CHECK( SUBSTR(Ename,1,1) BETWEEN 'A' AND 'Z')

AND Ename=UPPER(Ename)), Job Varchar2(15) Constraint Job\_Chk CHECK(Job IN('ANALYLT','CLERK','MANAGER','PRESIDENT','SALESMAN')), Hiredate date DEFAULT SYSDATE,

Sal Number(8,2) Constraint Sal\_NN NOT NULL

Constraint CHK\_Sal CHECK(Sal BETWEEN 1000 and 10000),

Comm number(8,2),

Deptno Number(2),

Constraint Tot\_Sal\_Chk CHECK(Sal+Comm< = 100000));

**Default Option**:

• If values are not provided for table column default will be considered.

• The options prevent NULL Values from entering the Columns, if a row is inserted without a value for a column.

• The DEFAULT value can be a literal, an expression or a SQL Function.

• The DEFAULT Expression must match the data type of the Column.

EX:

CREATE TABLE DEPT

(DEPTNO NUMBER(2),

DNAME VARCHAR2(14),

LOC VARCHAR2(13) default 'PUNE');

CREATE TABLE Emp

( Empno Number(4) Constraint Empno\_Pk PRIMARY KEY,

Ename Varchar2(20) Constraint ename\_NN NOT NULL,

Hiredate date DEFAULT SYSDATE,

Sal Number(8,2) Constraint Sal\_NN NOT NULL,

Comm number(8,2) default 0,

Deptno Number(2));

**Adding Constraint to a table**:

• A constraint can be added to a table at any time after the table was created by using by ALTER TABLE Statement, using **ADD** Clause.

**Syntax**: SQl>ALTER TABLE <Table Name>

ADD [Constraint <Constraint Name>]

Cons\_Type(Column\_Name,[ Column\_Name,..]);

**Guidelines:**

• The Constraint Name syntax is optional, but recommended.

• Table Constraints are applied to table if data previously placed in the table violated such constraints.

• We can ADD, DROP, ENABLE, or DISABLE a Constraint.

• NOT NULL, DEFAULT can be added to existing column by using the MODIFY Clause of the ALTER TABLE statement.

**Example**:

Sql>ALTER TABLE Emp ADD Constraint EMP\_EMPNO\_PK PRIMARY KEY(EMPNO);

Sql>ALTER TABLE Emp ADD Constraint EMP\_MGR\_FK FOREIGN KEY(MGR) REFERENCES Emp (EMPNO);

Sql>ALTER TABLE Emp ADD CONSTRAINT DEPT\_DEPTNO\_FK FOREIGN KEY(DEPTNO) REFERENCES Dept (DEPTNO);

Sql> ALTER TABLE Emp MODIFY Hiredate date default sysdate;

Sql> ALTER TABLE EMP MODIFY ENAME NOT NULL;

SQL>ALTER table emp modify job varchar2(9) constraint emp\_job\_nn not null;

**DROPPING Constraints**:

• To drop a constraint identify the constraint name from USER\_CONSTRAINTS and USER\_CONS\_COLUMNS Data dictionary views.

• The ALTER TABLE Statement is used with the DROP Clause.

• The CASCADE Option of the DROP Clause causes any dependent constraints also to be dropped.

• When a constraint is dropped, the constraint is no longer enforced and is no longer available in the data dictionary.

**Syntax**:

ALTER TABLE <Table Name>

DROP PRIMARY KEY/UNIQUE(Column)/

CONSTRAINT Constraint\_Name[CASCADE];

**Note**: When drop the PRIMARY KEY/UNIQUE Constraints the related INDEX will drop automatically.

Sql>ALTER TABLE Emp DROP PRIMARY KEY;

Sql>ALTER TABLE Dept DROP UNIQUE(Dname);

Sql> alter table emp drop constraint emp\_job\_nn;

Sql>SELECT index\_name from user\_indexes WHERE

table\_name = 'EMP';

**DISABLING Constraint**:

The constraint can be disabled without dropping it or recreating it.

The ALTER TABLE statement is used with the DISABLE Clause.

**Syntax**: ALTER TABLE <Table Name>

DISABLE CONSTRAINT <Constraint Name>[CASCADE];

**Guidelines**:

* The CASCADE clause disables dependent integrity constraints.

**Example**:

Sql> ALTER TABLE Emp DISABLE CONSTRAINT EMP\_EMPNO\_PK CASCADE;

Sql> ALTER TABLE Emp DISABLE CONSTRAINT EMP\_EMPNO\_PK;

**ENABLE Constraint**:

* The constraint can be enabled without dropping it or recreating it.
* The ALTER TABLE statement is used with the ENABLE Clause.

**Syntax:** ALTER TABLE <Table Name>

ENABLE CONSTRAINT <Constraint Name>;

• Enabling a Constraint applied to all the data in the table.

• When a UNIQUE or PRIMARY KEY Constraint is ENABLED, the

UNIQUE or PRIMARY KEY Index is automatically created.

Sql>ALTER TABLE Emp ENABLE CONSTRAINT EMP\_MGR\_FK;

**VIEWING Constraints**:

• To View all Constraints on table by Query the USER\_CONSTRAINTS table.

• The Codes that are revealed are....

* P-Primary
* U-Unique
* R-References
* C-Check & Not Null

Sql> SELECT owner,constraint\_name,constraint\_type

FROM user\_constraints WHERE

table\_name='EMP';

**VIEWING The Columns Associated With Constraints**:

* The Names of the columns that are involved in constraints can be known by querying the USER\_CONS\_COLUMNS Date Dictionary View.

Sql>SELECT constraint\_name,column\_name

FROM USER\_CONS\_COLUMNS WHERE

table\_name = 'EMP';

**Note**: When drop the table all corresponding Integrity Constrains will droped automatically.