# INTEGRITY CONSTRAINTS

The purpose INTEGRITY CONSTRAINTS in a relational DB is to maintain the accuracy of data.

Types of Integrity Constraints

1. **Domain constraints**

All the values under an attribute of a table must be from the same domain.

1. **Entity Integrity**

All Primary key values are valid and NOT NULL.

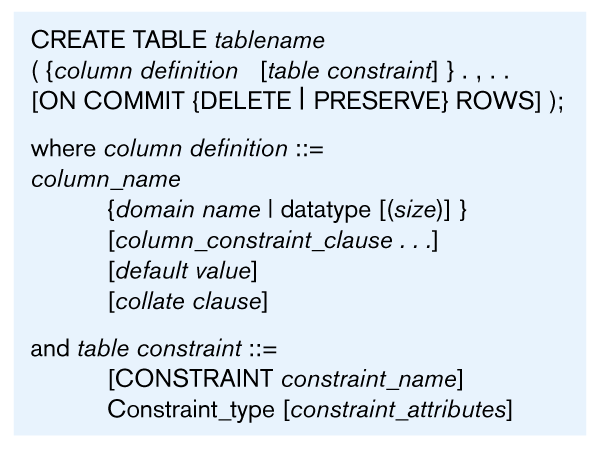
1. **Referential Integrity**

Each foreign key value in one relation (Table) must match t a primary key value in another Relation (Table).

1. **Action Assertions**

A statement of a constraint or control on the actions of the organisation.

# READ Page 192-195 (Ch5), Modern Database Management, Hoffer



**Hands on Example (Integrity constraints)**

Create the database with Entity Integrity and Referential Integrity constraints.

Use yourdatabase

CREATE TABLE student(

sno varchar(2) NOT NULL,

sname varchar(10) NULL,

address varchar(50) NULL,

CONSTRAINT stu\_sno\_pk PRIMARY KEY(sno))

Use yourdatabase

CREATE TABLE module(

mno varchar(10) NOT NULL,

marks int NULL,

sno varchar(2) NOT NULL,

CONSTRAINT module\_mno\_pk PRIMARY KEY (mno),

CONSTRAINT FK\_student\_module FOREIGN KEY(sno)

REFERENCES student(sno)ON DELETE CASCADE ON UPDATE CASCADE);

# Referential integrity constraints

You can define integrity constraints at the time you create a table. These constraints are rules that validate data entries by enforcing column-to-table and table-to-table relationships. They span all transactions that access the database and are automatically maintained by the system. CREATE TABLE supports the following integrity constraints:

A PRIMARY KEY is one or more columns whose collective contents are guaranteed to be unique. A PRIMARY KEY column must also define the NOT NULL attribute. A table can have only one primary key.

UNIQUE keys ensure that no two rows have the same value for a specified column or ordered set of columns. A unique column must also define the NOT NULL attribute. A table can have one or more UNIQUE keys. A UNIQUE key can be referenced by a FOREIGN KEY in another table.

Referential constraints (REFERENCES) ensure that values in the specified columns (known as the foreign key) are the same as values in the referenced UNIQUE or PRIMARY KEY columns in another table. The UNIQUE or PRIMARY KEY columns in the referenced table must be defined before the REFERENCES constraint is added to the secondary table. REFERENCES has ON DELETE and ON UPDATE clauses that define the action on the foreign key when the referenced primary key is updated or deleted. The values for ON UPDATE and ON DELETE are as follows:

## Action Effect on foreign key

**NO ACTION** [Default] The foreign key does not change. This may cause the primary key update or delete to fail due to referential integrity checks.

**CASCADE** The corresponding foreign key is updated or deleted as appropriate to the new value of the primary key.

**SET DEFAULT** Every column of the corresponding foreign key is set to its default value. If the default value of the foreign key is not found in the primary key, the update or delete on the primary key fails. The default value is the one in effect when the referential integrity constraint was defined. When the default for a foreign key column is changed after the referential integrity constraint is set up, the change does not have an effect on the default value used in the referential integrity constraint.

**SET NULL** Every column of the corresponding foreign key is set to NULL.