Daffodil International University

Chapter 2- Integrity Constraints

Course Code: CIS 322

Module Title: Advanced Database

Module Teacher: Nayeema Rahman

Relational Integrity

Two types of relational integrity:

- Entity Integrity
- Referential Integrity

Entity Integrity

Entity integrity concerns primary keys. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.

Referential Integrity

Referential integrity concerns foreign keys. The referential integrity rule states that any foreign key value can only be in one of two states. The usual state of affairs is that the foreign key value refers to a primary key value of some table in a database.

Domain

The set of predefined valid values of an attribute is known as domain.

For example: An attribute ‘Gender’ can have domain with a set of valid values for ‘Male’ or ‘Female’.

Propagation Constraint

Propagation constraints would need to be defined to enforce the rules as to what happens if data that is referenced elsewhere is altered or deleted.

Four types of propagation constraints are found:

- No action
- Cascade
- Set Default
- Set Null

Types of Propagation Constraint

- No action means the record in the table with the foreign key is left as it is.
- Cascade which means that any change (including a delete) is replicated in the table with the foreign key in it.
- Set Default, which means that the change in the parent table causes the record in the child table to be set to some sort of default.
- Set Null is similar to Set Default except that the table with the foreign key set to Null.

Super Key, Candidate Key

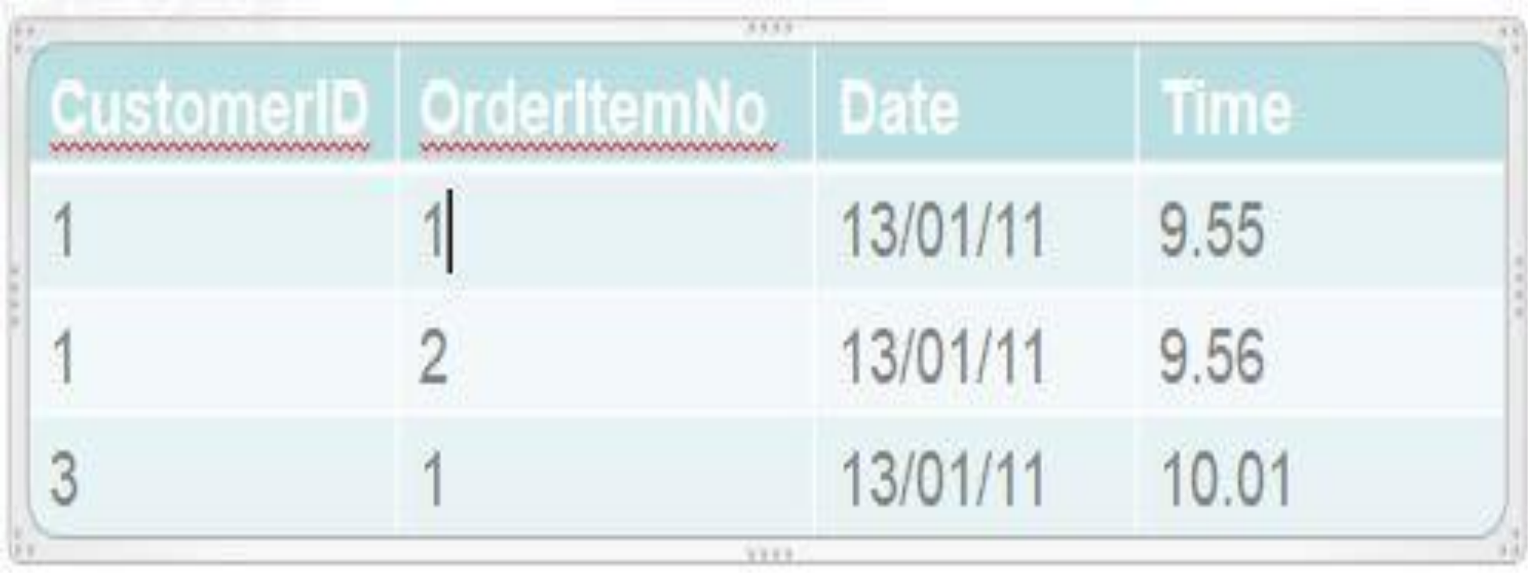
- **Super Key**

A set of attributes that can uniquely identifies a tuple. An attribute or set of attributes upon which all the other attributes are functionally dependent.

- **Candidate Key**

An attribute or set of attributes that is capable of uniquely identifying a row of a table.

Example of Super & Candidate Keys



CustomerID	OrderItemNo	Date	Time
1	1	13/01/11	9.55
1	2	13/01/11	9.56
3	1	13/01/11	10.01

Possible candidate keys?

Functional Dependency

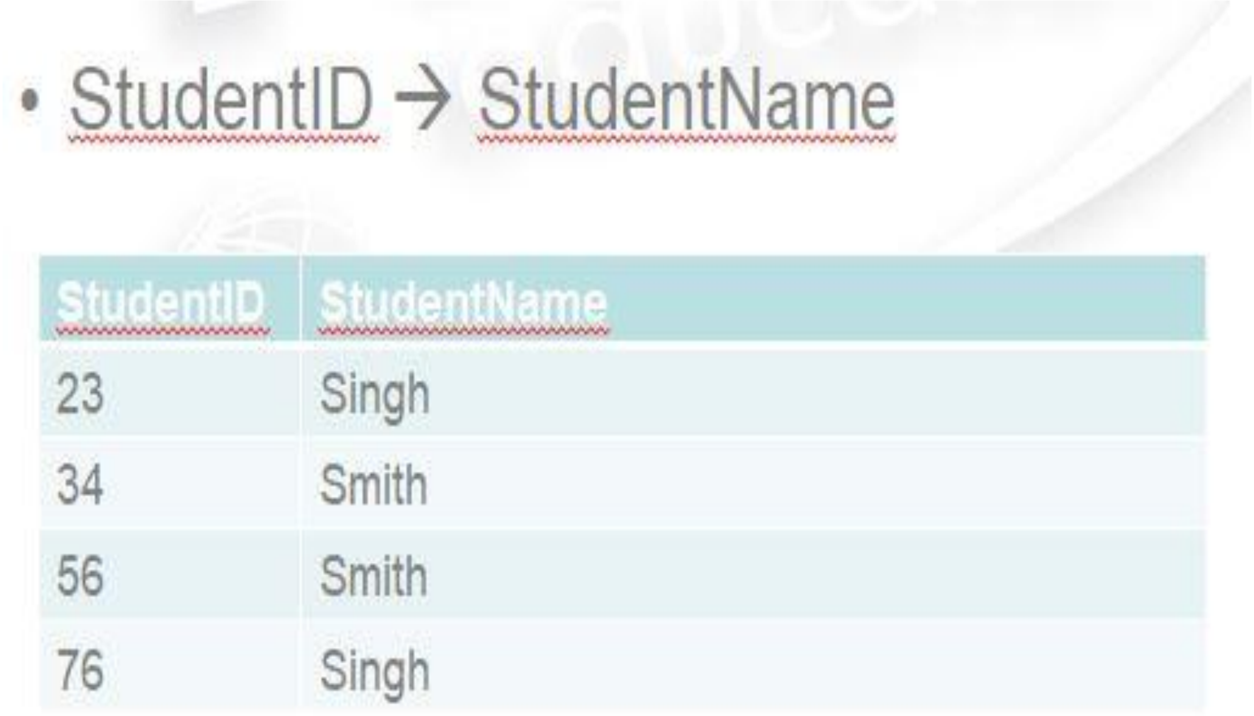
A->B

Within a relation if it is said that 'A determines B' then this means that if you know the value of 'A' then you will know the value of 'B'.

So A functionally determines B. Or B is functionally determined by A.

Note that the reverse is not true.

Example of Functional Dependency



- StudentID → StudentName

StudentID	StudentName
23	Singh
34	Smith
56	Smith
76	Singh