**Lesson 16**

**Objectives**

* Relational Model
  + Relation
  + Fields/Column
  + Domain
  + Tuple/Record
  + Degree
  + Cardinality
  + Relational Database
  + Properties of Relation
  + Foreign Key
  + Referential Integrity

**Relational Data Model**

A data model which is represented in the form of relations.

**Correspondence with E-R Model**

* Relations (tables) correspond with entity types and with many-to-many relationship types
* Rows correspond with entity instances and with many-to-many relationship instances
* Columns correspond with attributes

**Relation**

A relation is a name table with column and rows (only apply to logical structure). Physically there is no structure of table in memory.

In relational model; relation is an alternative of entities (ER model).

Relation can be represented in following two ways

1.

Product

|  |  |  |
| --- | --- | --- |
| ProductId | Product name | Price |

Journey

|  |  |
| --- | --- |
| JourneyId | Journey Name |

2.

Product(ProductID, Product name, Price)

Vendor(VendorID, Vendor Name)

**NOTE**: The word ***relation*** (in relational database) is NOT the same as the word ***relationship*** (in E-R model)

**Field/Column**

In relational model; field is an alternative of attributes (ER model). It is also called column. ProductId, Product name and Price are fields/columns of relation Product.

**Domain**

Domain is a set of allowable values for field/column. For example; allowable values for Price are number and for Product name is string.

**Degree**

Degree is a no. of fields/Column in a relation. Degree of Product and Journey is 3 and respectively.

**Records/Rows/Tuples**

Each row in a relation is called row/tuple. Row is an alternative of instance (ER model)

**Cardinality**

Cardinality is a no. of rows/records in a relation.

Cardinality of relation given below is 4.

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Name | Age | Address |
| 1 | Khurm | 23 | Multan |
| 2 | Ali | 19 | Islamabad |
| 3 | Hader | 17 | Lahore |
| 4 | Maria | 28 | rawalpindi |

**Properties of Relation**

* Relation name is distinct from all other relation names in database
* Each cell contains exotically one atomic value
* Each attribute has distinct name within a relation
* Values of attributes are from same domain
* Each type is distinct
* Order of columns has no significance
* Order of rows or topple has no significance

**Primary Key**

Concept of Primary Key is similar with ER model

**Foreign Key**

An attribute or set of attribute that serve as a primary key in one relation while exist as a non key or part of primary key in other relation with in a database

**Travel Agency**

|  |  |
| --- | --- |
| Agency Id | Agency Name |

**Travel Agent**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Agent Id | Agent Name | Agent Address | Cell no | NIC# | Joining Date | Agency Id |

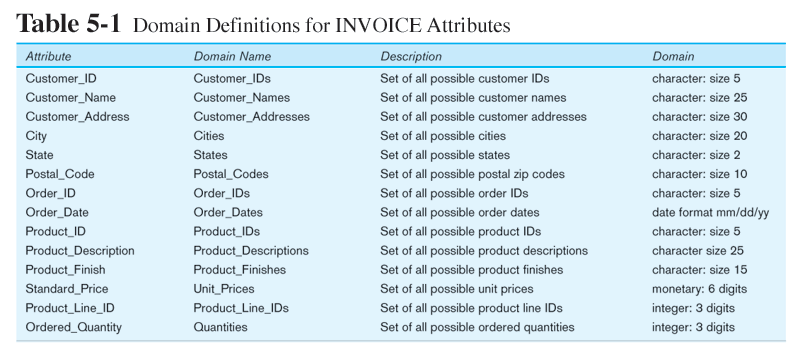
**Integrity Constraints**

Restriction on relation to ensure integrity of database is referred as integrity constraints. There are following type of integrity constraints.

* Domain Constraint
* Entity Integrity
* Action Assertions
* Referential integrity

**Domain Constraint**

Allowable values for a field/column (attribute) as given below



**Entity Integrity**

No primary key attribute may be null. All primary key fields **MUST** have data

**Action Assertions**

Business rules. Recall your assignment

**Referential Integrity**

Rule that states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of the one side. (Or the foreign key can be null)

For example: Delete Rules

* Restrict – don’t allow delete of “parent” side if related rows exist in “dependent” side
* Cascade – automatically delete “dependent” side rows that correspond with the “parent” side row to be deleted
* Set-to-Null – set the foreign key in the dependent side to null if deleting from the parent side 🡪 not allowed for weak entities