

TABLE 3.2 Summary of Basic Relational Database Constraints

Constraint	Description
Naming a Relation	Each relation in a relational schema must have a different name.
Naming a Column	Within a relation, each column must have a different name.
Row Uniqueness	Within a relation, each row must be unique.
Single-Valued Column Values	Within a relation, in each row, each value in each column must be single valued.
Domain Constraint	Within a relation, all values in each column must be from the same predefined domain.
Order of Columns	Within a relation, the order of columns is irrelevant.
Order of Rows	Within a relation, the order of rows is irrelevant.
Primary Key Constraint	Each relation must have a primary key, which is a column (or a set of columns) whose value is unique for each row.
Entity Integrity Constraint	No primary key column can have null values.
Foreign Key	A foreign key is a column in a relation that refers to a primary key column in another (referred) relation.
Referential Integrity Constraint	In each row of a relation containing a foreign key, the value of the foreign key EITHER matches one of the values in the primary key column of the referred relation OR the value of the foreign key is null (unfilled).

TABLE 3.3 Summary of Basic ER to Relational Schema Mapping Rules

ER Construct	Mapping Rule
Regular Entity	Becomes a relation.
Regular Attribute	Becomes a column in a relation.
Unique Attribute	One unique attribute of the entity becomes a primary key. If there are other unique attributes they are marked as unique (but not as the primary key).
Composite Attribute	Only components of the composite attribute are mapped as columns of a relation (the composite attribute itself is not mapped).
Composite Unique Attribute	Only its components are mapped. The components become a composite primary key only if there are no single unique attributes in the entity. Otherwise, the components are marked as compositely unique (but not as the primary key).
Multivalued Attribute	Becomes a separate relation with a composite primary key. The composite primary key is composed of a column representing the multivalued attribute and the foreign key referring to the primary key of the relation representing the entity that contained the multivalued attribute.
Derived Attribute	Not mapped.
Optional Attribute	Becomes a column, marked as optional.

TABLE 3.3 Summary of Basic ER to Relational Schema Mapping Rules (*Continued*)

ER Construct	Mapping Rule
1:M Binary Relationship	The relation mapped from the entity on the M side has a foreign key that corresponds to the primary key of the relation mapped from the 1 side.
M:N Binary Relationship	Becomes a new relation with two foreign keys, corresponding to the primary keys of the two relations representing the two entities involved in the M:N relationship. The two foreign keys form the composite primary key of the new relation representing the M:N relationship. If it exists, an attribute of the relationship becomes a column in the new relation.
1:M Unary Relationship	The relation mapped from an entity involved in the 1:M unary relationship includes a foreign key that corresponds to its own primary key.
M:N Unary Relationship	Becomes a new relation with two foreign keys, both of them corresponding to the primary key of the relation representing the entity involved in the unary M:N relationship. Each of the foreign keys is used as a part of the composite primary key of the new relation.
Associative Entity	Same rule as with mapping an M:N relationship. A new relation is created with foreign keys corresponding to the primary keys of the relations representing the entities involved in the relationships with the associative entity.
Weak Entity	Becomes a new relation with a foreign key corresponding to the primary key of the relation representing the owner entity. The combination of the column mapped from the partial key and the foreign key from the owner becomes a composite primary key (if there is no partial key, the foreign key alone becomes the primary key).
Ternary Relationship	Same rule as with mapping an associative entity.