**Synonyms and Cardinality**

Table of Contents

[Synonyms 2](#_Toc83821328)

[Cardinality 3](#_Toc83821329)

[One-to-One Relationships 3](#_Toc83821330)

[One-to-Many Relationships 4](#_Toc83821331)

[Many-to-Many Relationship 5](#_Toc83821332)

## Synonyms

**Synonyms** are layers in a database that create **abstraction**. This allows us to use **local names** for a whole bunch of things.

Image we have a server, server1, where we have a schema, abc, in which we have a table, employee. If we wanted to refer to this table from a remote location, such as from system, we would have to write a command like server1.abc.employee. We had to expose all the names to the end-user.

Instead, we can create a synonym that can be used to refer to the same table remotely.

CREATE PUBLIC SYNONYM employee\_table FOR server1.abc.employee;

SQL

Anything and everything that we could have done with the actual server or schema or table, we can do with this synonym.

## Cardinality

**Cardinality** defines the **degree** of the relationship between two tables. These can be:

1. , where each entry from the first table is related to exactly one entry from the other table,
2. , where each entry from the first table is related to multiple entries from the other table or
3. , where multiple entries from the first table are related to multiple entries from the other table.

### One-to-One Relationships

To establish a relationship, we need to create a **foreign key** from one table to the **primary key** of the other, with the additional constraint that the foreign key be **unique**.

CREATE TABLE t1 (  
 t1\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t1 PRIMARY KEY (t1\_id)  
);  
  
CREATE TABLE t2 (  
 t2\_id NUMBER(9, 0),  
 t1\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t2 PRIMARY KEY (t2\_id),  
 CONSTRAINT unique\_t1 UNIQUE (t1\_id),  
 CONSTRAINT fk\_t1 FOREIGN KEY (t1\_id) REFERENCES t1 (t1\_id)  
);

SQL

### One-to-Many Relationships

A relationship is easier to establish than an relationship, since it is done in the exact same way, except the **unique** constraint is not placed.

CREATE TABLE t1 (  
 t1\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t1 PRIMARY KEY (t1\_id)  
);  
  
CREATE TABLE t2 (  
 t2\_id NUMBER(9, 0),  
 t1\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t2 PRIMARY KEY (t2\_id),  
 CONSTRAINT fk\_t1 FOREIGN KEY (t1\_id) REFERENCES t1 (t1\_id)  
);

SQL

### Many-to-Many Relationship

Establishing a relationship requires using a junction table that contains the **primary keys** of both the tables being related. Both of these are also **foreign keys**.

CREATE TABLE t1 (  
 t1\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t1 PRIMARY KEY (t1\_id)  
);  
  
CREATE TABLE t2 (  
 t2\_id NUMBER(9, 0),  
 CONSTRAINT pk\_t2 PRIMARY KEY (t2\_id),   
);

CREATE TABLE junction (  
 t1\_id NUMBER(9, 0),  
 t2\_id NUMBER(9, 0),  
 CONSTRAINT pk\_junction PRIMARY KEY (t1\_id, t2\_id),  
 CONSTRAINT fk\_t1 FOREIGN KEY (t1\_id) REFERENCES t1 (t1\_id),  
 CONSTRAINT fk\_t2 FOREIGN KEY (t2\_id) REFERENCES t2 (t2\_id)  
);

SQL