

Created by Ahsan Arif

## Foreign Keys

---

A **foreign key** means that values in one table must also appear in another table.

The referenced table is called the **parent table** while the table with the foreign key is called the **child table**. The foreign key in the child table will generally reference a primary key in the parent table.

A foreign key can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.

### Using a CREATE TABLE statement

The syntax for creating a foreign key using a CREATE TABLE statement is:

```
CREATE TABLE table_name
(column1 datatype null/not null,
column2 datatype null/not null,
...
CONSTRAINT fk_column
  FOREIGN KEY (column1, column2, ... column_n)
  REFERENCES parent_table (column1, column2, ... column_n)
);
```

For example:

```
CREATE TABLE supplier
( supplier_id    numeric(10) not null,
  supplier_name varchar2(50) not null,
  contact_name  varchar2(50),
  CONSTRAINT supplier_pk PRIMARY KEY
  (supplier_id)
);
```

```
CREATE TABLE products
( product_id    numeric(10) not null,
  supplier_id   numeric(10) not null,
  CONSTRAINT fk_supplier
  FOREIGN KEY (supplier_id)
  REFERENCES supplier(supplier_id)
```

);

In this example, we've created a primary key on the supplier table called *supplier\_pk*. It consists of only one field - the *supplier\_id* field. Then we've created a foreign key called *fk\_supplier* on the products table that references the supplier table based on the *supplier\_id* field.

We could also create a foreign key with more than one field as in the example below:

```
CREATE TABLE supplier
( supplier_id    numeric(10) not null,
  supplier_name varchar2(50) not null,
  contact_name  varchar2(50),
  CONSTRAINT supplier_pk PRIMARY KEY (supplier_id,
  supplier_name)
);
```

```
CREATE TABLE products
( product_id    numeric(10) not null,
  supplier_id    numeric(10) not null,
  supplier_name  varchar2(50) not null,
  CONSTRAINT fk_supplier_comp
  FOREIGN KEY (supplier_id, supplier_name)
  REFERENCES supplier(supplier_id, supplier_name)
);
```

In this example, our foreign key called *fk\_foreign\_comp* references the supplier table based on two fields - the *supplier\_id* and *supplier\_name* fields.

## Using an ALTER TABLE statement

The syntax for creating a foreign key in an ALTER TABLE statement is:

```
ALTER TABLE table_name
add CONSTRAINT constraint_name
  FOREIGN KEY (column1, column2, ... column_n)
  REFERENCES parent_table (column1, column2, ... column_n);
```

For example:

```
ALTER TABLE products
add CONSTRAINT fk_supplier
  FOREIGN KEY (supplier_id)
  REFERENCES supplier(supplier_id);
```

In this example, we've created a foreign key called *fk\_supplier* that references the supplier table based on the `supplier_id` field.

We could also create a foreign key with more than one field as in the example below:

```
ALTER TABLE products
add CONSTRAINT fk_supplier
FOREIGN KEY (supplier_id, supplier_name)
REFERENCES supplier(supplier_id, supplier_name);
```