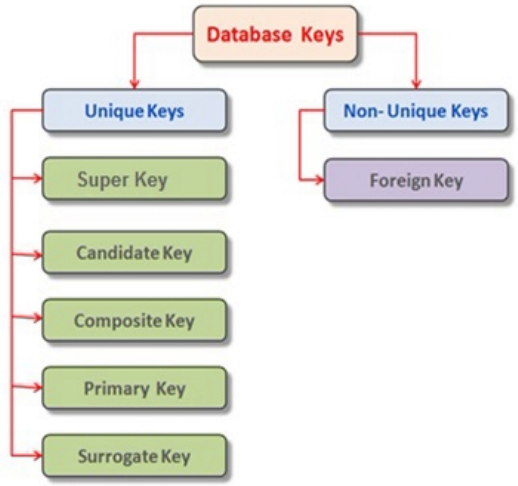
Relational DB & SQL - C11

Basic Concepts of the Relational Model

To do: Go through the activity to the end

Key Constraints

There are several types of keys, such as Candidate key, Composite key, Primary key (PK), Secondary key, Alternate key and Foreign key (FK).



Candidate key

A candidate key is a simple or composite key that is unique and minimal. It is unique because no two rows in a table may have the same value at any time. It is minimal because every column is necessary in order to attain uniqueness.

Composite key

A composite key is composed of two or more attributes, but it must be minimal.

Primary key

The primary key (PK) is a candidate key that is selected by the database designer to be used as an identifying mechanism for the whole entity set. It must uniquely identify tuples in a table and not be null. The primary key is indicated in the ER model by underlining the attribute.

A primary key is a column or a group of columns that uniquely identifies each row in a table. You create a primary key for a table by using the PRIMARY KEY constraint.

> 💡 **Tips:**
> - In case the primary key has two or more columns, you must use the PRIMARY KEY constraint as a table constraint.
> - Each table can contain only one primary key.
> - SQL Server automatically sets the `NOT NULL` constraint for all the primary key columns.
> - SQL Server also automatically creates a unique clustered index (or a non-clustered index if specified as such) when you create a primary key.

Secondary key

A secondary key is an attribute used strictly for retrieval purposes (can be composite).

Alternate key

Alternate keys are all candidate keys not chosen as the primary key.

Foreign key

A foreign key (FK) is an attribute in a table that references the primary key in another table. FK can be null. Both foreign and primary keys must be of the same data type.

The `FOREIGN KEY` (FK) constraint defines a column, or combination of columns, whose values match the `PRIMARY KEY` (PK) of another table.

- Values in an FK are automatically updated when the PK values in the associated table are updated/changed.
- FK constraints must reference PK or the `UNIQUE` constraint of another table.
- The number of columns for FK must be same as PK or `UNIQUE` constraint.
- If the `WITH NOCHECK` option is used, the FK constraint will not validate existing data in a table.
- No index is created on the columns that participate in an FK constraint.

In this example, the field id in the employee table is a FK to the field id in the departments table.

```
1  CREATE TABLE employee
2  (
3      id BIGINT NOT NULL,
4      name VARCHAR(20) NULL,
5    CONSTRAINT foreignkey_1 FOREIGN KEY (id) REFERENCES
         departments(id)
6  );
```

Previous Next