

# SQL

## Constraints

## SQL PRIMARY KEY Constraint

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID) );
```

## Another way

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT PK_Person PRIMARY KEY (ID,LastName) );
```

# Contd...

## Add Primary Key:

```
ALTER TABLE Persons
```

```
ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);
```

**Or**

```
ALTER TABLE Persons
```

```
ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);
```

## Drop Primary Key:

```
ALTER TABLE Persons DROP PRIMARY KEY;
```

## FOREIGN KEY

CREATE TABLE Orders (

OrderID int NOT NULL,

OrderNumber int NOT NULL,

PersonID int,

PRIMARY KEY (OrderID),

FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)

);

## SQL FOREIGN KEY on ALTER TABLE

To create a FOREIGN KEY constraint on the "PersonID" column when the "Orders" table is already created, use the following SQL:

**Syntax:** ALTER TABLE table\_name

ADD CONSTRAINT constraint\_name

FOREIGN KEY (foreign\_key\_name,...)

REFERENCES parent\_table(column\_name,...);

**Example:** ALTER TABLE Orders

ADD CONSTRAINT FK FOREIGN KEY (PersonID) REFERENCES  
Persons(PersonID);

## **DROP a FOREIGN KEY Constraint**

**Syntax:** ALTER TABLE table\_name

DROP FOREIGN KEY constraint\_name;

### **Example:**

ALTER TABLE Orders

DROP FOREIGN KEY FK;

## AUTO INCREMENT Field

### Syntax for MySQL:

```
CREATE TABLE Persons (  
    Personid int NOT NULL AUTO_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255), Age int,  
    PRIMARY KEY (Personid) );
```



## Example:

```
CREATE TABLE animals (  
    id MEDIUMINT NOT NULL AUTO_INCREMENT,  
    name CHAR(30) NOT NULL,  
    PRIMARY KEY (id) );  
  
INSERT INTO animals (name) VALUES  
    ('dog'),('cat'),('penguin'),  
    ('lax'),('whale'),('ostrich');  
  
SELECT * FROM animals;
```

## CHECK on CREATE TABLE

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18) );
```

## OR CREATE TABLE Persons (

```
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int, CONSTRAINT CH  
    CHECK (Age>=18) );
```

## Contd....

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Persons ( ID int NOT NULL,  
  
    LastName varchar(255) NOT NULL,  
  
    FirstName varchar(255), Age int, City varchar(255),  
  
    CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Sandnes') );
```

## CHECK on ALTER TABLE

```
ALTER TABLE Persons
```

```
ADD CHECK (Age>=18);
```

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

```
ALTER TABLE Persons
```

```
ADD CONSTRAINT CHK_PersonAge CHECK (Age>=18 AND City='Sandnes');
```

## Drop Check constraint

```
ALTER TABLE Persons
```

```
DROP CHECK CH;
```

```
ALTER TABLE Persons
```

```
DROP CHECK CHK_PersonAge;
```

## DEFAULT on CREATE TABLE

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes' );
```

## **DEFAULT on ALTER TABLE**

**ALTER TABLE** Persons

**ALTER** City **SET DEFAULT** 'Sandnes';

## **DROP a DEFAULT Constraint**

**ALTER TABLE** Persons

**ALTER** City **DROP DEFAULT**;

## CREATE INDEX Syntax

- Creates an index on a table. Duplicate values are allowed:

CREATE INDEX *index\_name*

ON *table\_name* (*column1*, *column2*, ...);

## CREATE UNIQUE INDEX Syntax

- Creates a unique index on a table. Duplicate values are not allowed:

CREATE UNIQUE INDEX *index\_name*

ON *table\_name* (*column1*, *column2*, ...);



# Example

```
CREATE INDEX idx_lastname ON Persons (LastName);
```

If you want to create an index on a combination of columns, you can list the column names within the parentheses, separated by commas:

```
CREATE INDEX idx_pname ON Persons (LastName, FirstName);
```

## **DROP INDEX Statement**

```
ALTER TABLE table_name DROP INDEX index_name;
```

## SQL Date Data Types

**MySQL** comes with the following data types for storing a date or a date/time value in the database:

- DATE - format YYYY-MM-DD
- DATETIME - format: YYYY-MM-DD HH:MI:SS
- YEAR - format YYYY or YY