Click to prove you're human

By Cristian G. GuaschUpdated: 03/03/2410 min readNavigating the world of SQL can feel like decoding a complex puzzle, especially when it comes to understanding constraints. Among these, the DEFAULT constraint often flies under the radar, yet its a powerhouse for ensuring data integrity and simplifying database management. Its like having a safety net, ensuring that your data columns have a fallback value, making your database more robust and error-proof.Ive spent years tinkering with databases, and Ive come to appreciate the subtle art of using the SQL DEFAULT constraint effectively. Its not just about setting default values; its about crafting smarter, more resilient databases. Whether youre a beginner or looking to polish your skills, Ill guide you through the ins and outs of using this constraint to its full potential. Lets dive into the world of SQL together and unlock the secrets of the DEFAULT constraint.Understanding the SQL Default ConstraintWhen I first started with databases, grasping the SQL DEFAULT constraints utility was a game-changer for me. Its a brilliant tool that assigns a fallback value to a column when no value is specified during data entry. The essence of the DEFAULT constraint lies in its ability to maintain data integrity and ensure consistency across the database without additional effort from users. Implementation is straightforward. Lets say were creating a table for storing customer information, and we want to ensure that every new customer has a membership status. We might not have this information at the time of data entry, so we set a DEFAULT value.CREATE TABLE Customers (ID int NOT NULL,Name varchar(255) NOT NULL,MembershipStatus varchar(50) DEFAULT 'Pending');In this example, if a new customers MembershipStatus isnt specified, it automatically sets to Pending. Its versatile. You can also alter existing tables to add a DEFAULT constraint:ALTER TABLE CustomersADD CONSTRAINT df_MembershipStatusDEFAULT 'Pending' FOR MembershipStatus;This command adjusts the Customers table to ensure future entries have a default MembershipStatus of Pending if not specified. A common pitfall Ive noticed is forgetting that the DEFAULT constraint doesnt override explicit NULL values. If you insert a record with a NULL value for a column with a DEFAULT constraint, the NULL value takes precedence. To avoid this, always couple DEFAULT constraints with NOT NULL, if the column must always have a value.CREATE TABLE Orders (OrderID int NOT NULL,OrderDate date DEFAULT CURRENT_DATE, -- Automatically uses the current datePaymentReceived boolean NOT NULL DEFAULT FALSE);Here, OrderDate will default to the current date if not specified, and PaymentReceived defaults to FALSE, reinforcing the need to explicitly manage data entries to maintain database integrity.By understanding and applying the SQL DEFAULT constraint correctly, Ive been able to design more robust and error-proof databases. Its about ensuring that even when explicit values are missing, the database still behaves predictably, which is crucial for data integrity and operational continuity.Benefits of Using Default ConstraintsIn my experience, integrating default constraints into database design brings numerous advantages, especially when it comes to data integrity and system efficiency. These benefits have played a crucial role in streamlining my database management practices, ensuring smoother operations and more reliable data handling.Firstly, default constraints assure that a database remains populated with logical data, even when explicit values arent provided. This is particularly beneficial for fields like creation_date, where automatically recording the time of data entry is crucial. Without a default value, these fields could end up null, leading to potential inconsistencies or extra checks needed in application logic.Moreover, using default constraints significantly reduces the amount of error-prone manual data entry. By setting sensible defaults, Ive noticed a reduction in data entry errors, which in turn, enhances the quality of the data stored. For instance, setting a default status for new entries in a task management system ensures every task is trackable from the moment its created.Heres how to implement a default constraint in SQL:CREATE TABLE Employees (ID int NOT NULL,Name varchar(255) NOT NULL,HireDate date DEFAULT GETDATE());In this example, HireDate automatically gets the current date thanks to the DEFAULT GETDATE() constraint, ensuring no employee record is added without a hire date. While default constraints are invaluable, theyre not without pitfalls. A common mistake Ive encountered is neglecting to set defaults for boolean fields, leading to ambiguity between false and null values. For clarity, its best to explicitly define a default:ALTER TABLE Users ADD IsActive bit DEFAULT 1;This approach explicitly states that new users are active by default, eliminating any confusion between unassigned and false values.Variations in how defaults are set can also lead to surprises. For instance:CREATE TABLE Orders (ID int NOT NULL,OrderDate date NOT NULL DEFAULT GETDATE(),ShipDate date DEFAULT NULL);Here, OrderDate gets the current date by default, while ShipDate intentionally allows nulls, indicating an order might not yet be scheduled for shipment.Understanding these nuances and applying default constraints correctly has empowered me to build more resilient and user-friendly databases, streamlining operations and reinforcing data integrity. Syntax and Usage of Default Constraint in SQLWhen it comes to enhancing the integrity and usability of your database, understanding the syntax and usage of the default constraint in SQL is crucial. Ive found through experience that leveraging default constraints not only streamlines data entry but also maintains consistency across your database. Lets dive into the syntax and see how its implemented with practical code examples.To add a default constraint to an existing table, the syntax is as follows:ALTER TABLE table_nameADD CONSTRAINT constraint_nameDEFAULT default_value FOR column_name;For a new table, you can specify the default constraint directly in the column definition:CREATE TABLE table_name (column_name data_type DEFAULT default_value,...);Heres how I typically use it in real-world scenarios. Lets say we have an Employees table, and we want to ensure that if no hire date is specified, it automatically defaults to the current date. Heres how to do it:ALTER TABLE EmployeesADD CONSTRAINT df_HireDateDEFAULT GETDATE() FOR HireDate;Moreover, implementing a default constraint for a boolean field often trips up many developers. A common mistake is forgetting to set a default for such fields, leading to null values where true or false is expected. Heres how to avoid that:CREATE TABLE UserNotifications (NotificationEnabled BIT DEFAULT 1,...);By setting NotificationEnabled to 1, I ensure that users are opted into notifications by default, aligning with the principle of opting out rather than opting in.In working with various databases, Ive observed a variety of approaches to setting defaults. Some developers prefer setting defaults directly in application logic, but this can lead to inconsistencies if multiple applications interact with the same database. Therefore, defining defaults at the database level is a practice I strongly recommend.Throughout these examples, its apparent how default constraints can significantly impact data quality and application logic. By understanding and utilizing these SQL features, we can design more robust, efficient databases.Working with Default Constraints in Different Database SystemsNavigating through the use of default constraints can vary significantly across different database systems. Each system has its nuances, and knowing how to implement these constraints effectively requires understanding these variations. Lets delve into some examples and common mistakes to watch out for.In MySQL, adding a default constraint is straightforward. However, its important to remember that MySQL does not explicitly use the CONSTRAINT keyword for defaults. Heres how youd set a default value for a status column:ALTER TABLE OrdersMODIFY COLUMN status VARCHAR(255) DEFAULT 'Pending';One common mistake in MySQL involves forgetting to specify the column type with the MODIFY COLUMN clause when adding a default value. This omission can lead to unexpected errors or undesired table alterations.Moving over to SQL Server, the syntax becomes a bit different. SQL Server allows for more explicit naming of constraints, which can be beneficial for database maintenance and management. Heres how you can add a named default constraint:ALTER TABLE OrdersADD CONSTRAINT df_statusDEFAULT 'Pending' FOR status;A frequent oversight in SQL Server is neglecting to name the constraint. While its not mandatory, unnamed constraints receive system-generated names that can be cryptic and challenging to manage, especially when you need to modify or delete them.PostgreSQL offers a syntax similar to SQL Server but with slight differences. For instance, when adding a default constraint, you normally wouldnt need to worry about naming it, as the focus is often on the action rather than the constraints identity:ALTER TABLE OrdersALTER COLUMN status SET DEFAULT 'Pending';A common pitfall in PostgreSQL is attempting to add a default value to a column without specifying the SET DEFAULT clause, leading to confusion and potential syntax errors.Across all these database systems, its vital to understand the specific syntax and best practices. By avoiding common mistakes and leveraging the nuances of each system, you can ensure that your database utilizes default constraints effectively, enhancing data integrity and streamlining operations.Practical Examples and Best PracticesWhen diving into the world of SQL, mastering the use of default constraints can significantly streamline your database management processes. Lets break down some practical examples across MySQL, SQL Server, and PostgreSQL to enhance your understanding and application of this feature.Starting with MySQL, setting a default value for a column is straightforward. Suppose we have a table named Users and we want to ensure that any new user added has a default role of guest. The SQL statement would look like this:ALTER TABLE Users ADD COLUMN role VARCHAR(50) DEFAULT 'guest';A common mistake here is forgetting to specify the column type, which results in an error. Always remember the importance of clearly defining your data types.Moving on to SQL Server, we can assign a default constraint while creating a new table. For example, if were setting up a table for product information and we want all products to have a default stock quantity of 10, the command would be:CREATE TABLE Products (ProductID int NOT NULL,ProductName varchar(255) NOT NULL,StockQuantity int NOT NULL DEFAULT 10);In SQL Server, its a best practice to name your constraints. This aids in easy identification and management. Heres how you could do it for the previous example:ALTER TABLE ProductsADD CONSTRAINT DF_StockQuantity DEFAULT 10 FOR StockQuantity;For PostgreSQL, the approach is quite similar, but with its nuances. Lets say were managing an events table and each new event is initially set as not featured. The SQL might look like this:ALTER TABLE Events ADD COLUMN isFeatured BOOLEAN DEFAULT FALSE;One crucial aspect in PostgreSQL is ensuring that if operations on the table require altering the default value, the SET DEFAULT clause is preferred. This ensures that any alterations adhere to the predefined defaults, maintaining data consistency.Through these examples, its clear that while the concepts remain consistent across database systems, syntax and best practices can vary. Whether its ensuring data types in MySQL, naming conventions in SQL Server, or utilizing the SET DEFAULT clause in PostgreSQL, being aware of these can save you from common pitfalls and streamline your database management efforts.ConclusionMastering the use of default constraints across MySQL, SQL Server, and PostgreSQL can significantly enhance your database management skills. By paying attention to the specific requirements and best practices of each database system, youll ensure data integrity and streamline your workflows. Whether its specifying column types in MySQL, naming constraints in SQL Server, or using the SET DEFAULT clause in PostgreSQL, these insights are invaluable. Remember, a well-managed database is the backbone of any robust application, and with these tips, youre well-equipped to handle any challenges that come your way. The SQL DEFAULT Constraint is used to specify the default value for a column of a table. We usually set default value while creating the table.The default values are treated as the column values if no values are provided while inserting the data, ensuring that the column will always have a value. We can specify default values for multiple columns in an SQL table. SyntaxFollowing is the syntax of the SQL DEFAULT Constraint CREATE TABLE table_name (column1 datatype DEFAULT default_value, column2 datatype DEFAULT default_value, column3 datatype, ... columnN datatype);ExampleIn the following query we are creating the CUSTOMERS table using the CREATE TABLE statement. Here, we are adding a default constraint to the columns NAME, AGE, ADDRESS, and SALARY CREATE TABLE CUSTOMERS (ID INT NOT NULL, NAME VARCHAR (20) NOT NULL DEFAULT 'Ramesh', AGE INT NOT NULL DEFAULT '32', ADDRESS CHAR (25) DEFAULT 'Ahmedabad', SALARY DECIMAL (18, 2) DEFAULT '2000', PRIMARY KEY (ID));Following query inserts values into this table using the INSERT statement INSERT INTO CUSTOMERS (ID) VALUES (1);INSERT INTO CUSTOMERS VALUES (2, 'Khilan', 25, 'Delhi', 1500.00);The table is created with default values in the NAME, AGE, ADDRESS, and SALARY columns for the first row as shown below IDNAMEAGEADDRESSSALARY1Ramesh32Ahmedabad2000.002Khilan25Delhi1500.00 While inserting data into a table, if the column names are not included in the INSERT query, to insert the default value into the record we need to pass "DEFAULT" as a value, as shown below INSERT INTO CUSTOMERS VALUES(3, 'Kaushik', DEFAULT, DEFAULT, 2000.00),(4, 'Chaitali', DEFAULT, DEFAULT, DEFAULT);The table obtained is as shown below IDNAMEAGEADDRESSSALARY1Ramesh32Ahmedabad2000.002Khilan25Delhi1500.003Kaushik32Ahmedabad2000.004Chaitali32Ahmedabad2000.00Adding Default Constraint to an Existing ColumnWe can also add default constraints to an existing column of a table using the ALTER TABLE statement. This allows us to modify the structure of existing table by specifying default values, ensuring data consistency in the database.SyntaxFollowing is the syntax for adding a default constraint to a column in an existing table ALTER TABLE table_nameALTER COLUMN column_name SET DEFAULT 'default_value';ExampleAssume we have created another table named BUYERS using the CREATE TABLE statement as shown below CREATE TABLE BUYERS (ID INT NOT NULL, NAME VARCHAR (20) NOT NULL, AGE INT NOT NULL, ADDRESS CHAR (25), SALARY DECIMAL (18, 2), PRIMARY KEY (ID));Following query, we adds a default constraint to the ADDRESS column of the BUYERS table.ALTER TABLE BUYERS ALTER ADDRESS SET DEFAULT 'Delhi';Following INSERT statement inserts a record into the BUYERS table by providing values to all the columns except ADDRESS INSERT INTO BUYERS (ID, NAME, AGE, SALARY) VALUES(01, 'Rahul', 27, 50000);VerificationAfter inserting the record if you retrieve it back, you can observe the default value ("Delhi") in the address column SELECT * FROM BUYERS WHERE ID = 01;OutputThe table obtained is as follows IDNAMEAGEADDRESSSALARY01Rahul27Delhi50000.00Dropping Default ConstraintWe can delete the default constraint from a table using the ALTER TABLE... DROP statement.SyntaxFollowing is the syntax to delete the default constraint from a table ALTER TABLE table_nameALTER COLUMN column_name DROP DEFAULT;ExampleIn here, we are removing the default constraint from the ADDRESS column of the CUSTOMERS table ALTER TABLE CUSTOMERS ALTER ADDRESS DROP DEFAULT;VerificationWe can verify the table details (structure) and check whether there is a default constraint or not using the following query DESC CUSTOMERS;The table obtained is as shown below FieldTypeNullKeyDefaultExtraIDintNOPRINULLNAMEvarchar(20)NORameshAGEintNO32ADDRESSchar(25)YESNULLSALARYdecimal(18,2)YES2000.00November 4, 2022SQL default constraint is specified at the time of table creation or can be added with SQL ALTER table. SQL default constraint is used to specify a default value for a column; the value will be inserted into all new records if a value for that particular column is not set.SQL ALTER TABLE statement is used to make changes in the existing SQL table. It is used to add, remove or modify columns and add or remove constraints from an existing table. SQL ALTER TABLE ADD COLUMN statement is used to add a new column in the existing SQL table. Syntax of SQL ALTER TABLE statementALTER TABLE table_nameADD column_name datatype column_constraint;Here,table_name: Specify table name in which column will be addedcolumn_name:Specify the name of a column to be added, with datatypecolumn_constraint:Specify constraint name to be applied on a column Syntax of SQL add column with default valueALTER TABLE table_nameADD column_name datatype DEFAULT default_value;Here,DEFAULT: Keyword used to specify default constraintdefault_value:Specify the default value for a column Examples of SQL add column with default valueConsider student result management database with three tables student, result, and subject to perform SQL add column with default value examplesStudent table Result table Subject table SQL ADD Single column of int datatype column with default valueExample 1: Write SQL query to add new column semester in existing result tablealter table tblresult add semester int default 3In the above query, SQL alter table statement is applied to add a new column semester of int type in result table with default constraintDefault value 3 has been specified with default constraint which will automatically add when a new record is inserted.OUTPUT:INSERT INTO tblresult(result_id, student_id, examname, examdate, subjectid, obtainmark, totalmarks, pecentage, grade, status)VALUES (1010, 108, 'sem3', '02-02-2000', 1, 76, 100, 76, 'A', 'pass')In the above SQL insert query, the value for the newly added column semester does not specify. When we execute the above query new record will be inserted and a default value for the semester column will be added automatically.SELECT result_id, student_id, examname, examdate, subjectid, obtainmark, totalmarks, pecentage, grade, status, semesterFROM tblresult SQL ADD Single Varchar datatype column with default valueIn SQL to add a new column of varchar datatype with a default value, we need to specify character length with Varchar type and also need to define a default value in the single quotation.Example 2: Write SQL query to add new column address of varchar type with character length 30 and default value as Althan, Surat in the student tablealter table tblstudent add address varchar(30) default 'althan, surat'In the above query, SQL alter table add column is used to add a new column address in the student tableThe default constraint is also specified with new column address with default value Althan,Surat'OUTPUT:INSERT INTO tblstudent(student_id, studentname, admissionno, admissiondate, enrollmentno, date_of_birth, email, city, class_id)VALUES(109, 'kiya', 30005, '09-09-2001', 'e163030303', '01-01-1992', 'kiya@gmail.com', 'Surat', 3);In the above SQL insert query, the value for the newly added column address does not specify. When we execute the above query new record will be inserted and a default value for the address column as althan, Surat will be added automatically.SELECT student_id, studentname, admissionno, admissiondate, enrollmentno, date_of_birth, email, city, class_id, address, age, phoneFROM tblstudent SQL Add multiple columns of int type with default valueWe can add multiple columns in a table using a single SQL alter table statement separated by comma(,). We need to specify each new column name with datatype and default value.Example 3: Write SQL query to add two new columns age and phone of int datatype in student tablealter table tblstudent add age int default 17, phone int default 98989In the above query, SQL alter table statement is used to alter existing table student by adding two new columns age and phone of int data typethe default constraint is also applied on both columns age with a default value 17 and phone with a default value as 98989OUTPUT:INSERT INTO tblstudent(student_id, studentname, admissionno, admissiondate, enrollmentno, date_of_birth, email, city, class_id)VALUES (111, 'riya', 30005, '09-09-2002', 'e163030313', '01-01-1992', 'riya@gmail.com', 'Surat', 3)In the above SQL insert query, values for recently added columns age and phone do not specify. When we execute the above query new record will be inserted and a default value for the age column as 17 and the phone column as 98989 will be added automatically.SELECT student_id, studentname, admissionno, admissiondate, enrollmentno, date_of_birth, email, city, class_id, address, age, phoneFROM tblstudent SQL Add multiple columns of varchar type with default valueExample 4: Write SQL query to add two columns exam_type and remark in result table with default value as external and presentalter table tblresult add exam_type varchar(20) default 'external', remark varchar(10) default 'present'In the above query, SQL alter table statement is used to alter existing table student by adding two new columns exam_type and remark of varchar datatypeThe default constraint is applied on both the columns exam_type and remarkOUTPUT:INSERT INTO tblresult (result_id, student_id, examname, examdate, subjectid, obtainmark, totalmarks, pecentage, grade, status, semester)VALUES (1021, 110, 'sem2', '02-02-2002', 2, 94, 100, 76, 'A', 'pass', 2)In the above SQL insert query, values for recently added columns exam_type and remark do not specify. When we execute the above query new record will be inserted and a default value for the exam_type column as external and remark column as present will be added automatically.SELECT result_id, student_id, examname, examdate, subjectid, obtainmark, totalmarks, pecentage, grade, status, semester, exam_type, remarkFROM tblresult SQL Add multiple columns with default and NOT NULL constraintIn SQL we can apply more than one constraint to a single column. We can apply default and NOT NULL constraints on a column separated by space. The NOT NULL constraint is used to enforce the user to enter a value in the specified column.Example 5: Write SQL query to add new column faculty_contact of varchar type with default value as 90909090 and NOT NULL constraint in the subject tablealter table tblsubject add faculty_number varchar(10) default '90909090' NOT NULL;In the above query, SQL alter table statement is used to alter subject table by adding new column faculty_number of varchar type.In the above query two constraints have been applied to faculty_number column. SQL default constraint to assign a default value as 90909090 and NOT NULL constraint.OUTPUT:INSERT INTO tblsubject (subjectid, facultname, subjectname, subjectcode)VALUES (4, 'ram', 'java', 1005)In the above SQL insert query, the value for the recently added column faculty_number does not specify. When we execute the above query new record will be inserted and a default value for faculty_number column as 90909090 will be added automatically.SELECT subjectid, facultname, subjectname, subjectcode, faculty_numberFROM tblsubject SummaryIn this article on SQL add column with default value, We have covered an overview of how to add a new column in an existing table, what is default constraint, the Syntax of SQL alter table add column, the syntax of SQL alter table add column with default constraint, also explain SQL add a single column with default constraint, SQL add multiple columns with default constraint with practical examples. ReferencesSQL alter tableSQL constraints Read MoreSQL specify default values for columns Summary: in this tutorial, you will learn you how to use the SQL DEFAULT constraint to insert a default value into a column.The DEFAULTconstraint inserts a default value into a column of a table when you insert a new row into the table without specifying the value for the column.There are two ways to create DEFAULTconstraints for columns:You can assign a DEFAULTconstraint to a column in the CREATE TABLE statement as the following statement:CREATE TABLE books (book_id INT NOT NULL PRIMARY KEY, title varchar(255) NOT NULL, pubdate date NOT NULL, isbn varchar(13) DEFAULT '1-84356-028-3', author_id INT NOT NULL)Code language: SQL (Structured Query Language) (sql)The isbncolumn in the bookstable accepts 1-84356-028-3 as the default value. If we insert a new row into the books table without specifying the value for ISBNcolumn, the database engine will insert the value 1-84356-028-3 into the isbncolumn. See the following INSERT statement that adds a new book to the bookstable:INSERT INTO books(title,pubdate,author_id)VALUES('SQL Tutorial','2010-01-01',1);Code language: SQL (Structured Query Language) (sql)We can query the bookstable by using SELECT statement to see what value has been inserted into the isbncolumn:SELECT * FROM books;Code language: SQL (Structured Query Language) (sql)The column ISBNstores the default value 1-84356-028-3 specified in the table definition.The DEFAULTconstraint accepts not only literal value but also a value returned by a function. For example, we assign today date as the default value for the pubdatecolumn by using the following statement:ALTER TABLE books ADD CONSTRAINT df_pubdateDEFAULT GETDATE() FOR pubdateCode language: SQL (Structured Query Language) (sql)To remove an existing DEFAULTconstraint, you also use the ALTER TABLE statement as follows:ALTER TABLE tableALTER COLUMN column DROP DEFAULT;Code language: SQL (Structured Query Language) (sql)For example, to remove the DEFAULTconstraint of the titlecolumn in the bookstable, you use the following statement:ALTER TABLE booksALTER COLUMN title DROP DEFAULT;Code language: SQL (Structured Query Language) (sql)In this tutorial, weve shown you how to use the SQL DEFAULT constraint to insert a default value into a column. SQL Server Max Int Value You can add a default value to a new OR existing column in SQL Server without dropping it. Specifying a default constraint should not cause any blocking. Only when setting a column to NULL OR NOT NULL will cause blocking. But to set the NULLABLE attribute, you need to use ALTER COLUMN instead.When adding this default value, its always best to specify the constraint name. If you dont the system will generate a random constraint name instead. This can wreak havoc because this default constraint name will not be the same name generated on your dev / stage / production systems. This is why its always best to specify the name.Add Default Value to Existing Column[cc lang=sql] Add default to existing column DateOfHire:ALTER TABLE [dbo].[Employees] ADD CONSTRAINT DF_Employees_DateOfHire DEFAULT (GETDATE()) FOR [DateOfHire] Add default value to existing column IsTerminatedALTER TABLE [dbo].[Employees] ADD CONSTRAINT DF_Employees_IsTerminated DEFAULT (0) FOR [IsTerminated][/cc]Add New Column with Default Value[cc lang=sql] Add new column DateOfHire with defaultALTER TABLE Employees ADD DateOfHire datetime CONSTRAINT DF_Employees_DateOfHire DEFAULT (GETDATE()) Add new column IsTerminated with default (no constraint name specified NOT RECOMMENDED)ALTER TABLE Employees ADD IsTerminated datetime DEFAULT (0)[/cc]Add Default Value with Create Table[cc lang=sql]CREATE TABLE [dbo].[Employees]([EmployeeID] [int] IDENTITY(1,1) NOT NULL, [FirstName] [varchar](50) NULL, [LastName] [varchar](50) NULL, [SSN] [varchar](9) NULL, Add default of zero [IsTerminated] [bit] NOT NULL CONSTRAINT DF_Employees_IsTerminated DEFAULT (0) , Add default of getdate() [DateAdded] [datetime] NULL CONSTRAINT DF_Employees_DateAdded DEFAULT (getdate()), [Comments] [varchar](255) NULL, [DateOfHire] [datetime] NULL)[/cc] Filed under DDL The DEFAULT constraint is used to set a default value for a column.The default value will be added to all new records, if no other value is specified. SQL DEFAULT on CREATE TABLEThe following SQL sets a DEFAULT value for the "City" column when the "Persons" table is created: My SQL / SQL Server / Oracle / MS Access:CREATE TABLE Persons(ID int NOT NULL,LastName varchar(255) NOT NULL,FirstName varchar(255),Age int,City varchar(255) DEFAULT 'Sandnes'); The DEFAULT constraint can also be used to insert system values, by using functions like GETDATE():CREATE TABLE Orders(ID int NOT NULL,OrderNumber int NOT NULL,OrderDate date DEFAULT GETDATE()); To create a DEFAULT constraint on the "City" column when the table is already created, use the following SQL:MySQL:ALTER TABLE PersonsALTER City SET DEFAULT 'Sandnes'; SQL Server:ALTER TABLE PersonsADD CONSTRAINT df_City DEFAULT 'Sandnes' FOR City; MS Access:ALTER TABLE PersonsALTER COLUMN City SET DEFAULT 'Sandnes'; Oracle:ALTER TABLE PersonsMODIFY City DEFAULT 'Sandnes'; DROP a DEFAULT ConstraintTo drop a DEFAULT constraint, use the following SQL:MySQL:ALTER TABLE PersonsALTER City DROP DEFAULT; SQL Server / Oracle / MS Access:ALTER TABLE PersonsALTER COLUMN City DROP DEFAULT; When using SQL Server, sometimes you need to modify an existing table. For the purposes of this article, say you want to add a DEFAULT constraint to an existing column. To add a DEFAULT constraint to an existing column, use the ALTER TABLE statement and specify the column and the specific constraint that you want to apply. Heres an example of adding a DEFAULT constraint to an existing column called TaskDescription: USE Solutions;ALTER TABLE Tasks ADD CONSTRAINT taskdesc_default DEFAULT 'TBA' FOR TaskDescription; GO In this case, we changed to the Solutions database first to ensure that we created the constraint against the correct database. We then went ahead and created a DEFAULT constraint calledtaskdesc_defaultfor the TaskDescription column, and set the default value to TBA. This means that, whenever a new row is created, if no value has been provided for the TaskDescription column, a default value of TBA will be inserted. You can check that the constraint has been created by running the following statement: USE Solutions;SELECT * FROM sys.default_constraints; This lists out all the default constraints for the Solutions database. Again, we changed to the correct database first. If your database has a lot of constraints, you can always narrow it down to just the constraint youre interested in: USE Solutions;SELECT * FROM sys.default_constraintsWHERE name = 'taskdesc_default';

Largest dinosaur natural history museum. How many dinosaurs are there in the natural history museum. Dinosaurs natural history museum london. What dinosaurs are in the museum of natural history. Naturalis dino. Naturalis dinosaurussen.