Relational Database Management System, DML, DDL

.NET

Chinook DB Seeder Code

https://raw.githubusercontent.com/2002-feb24-net/trainer-code/master/2-sql/Chinook_SqlServer.sql

---

Above ^^^

A software system used to maintain relational databases is a **Relational Database Management System (RDBMS)**. Many relational database systems use ***SQL*** (***Structured Query Language***) for querying and maintaining the database.

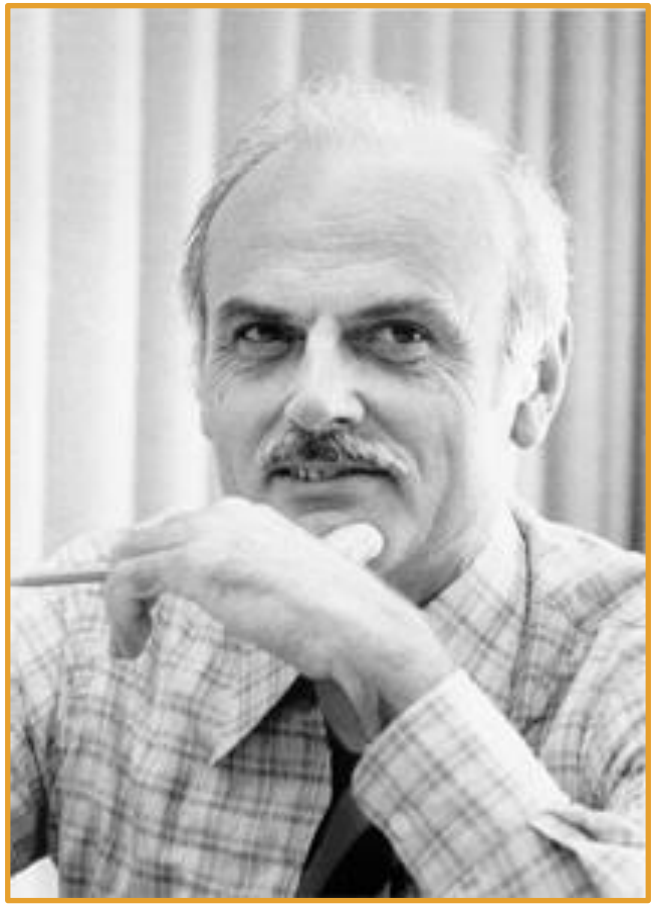
HTTPS://DOCS.MICROSOFT.COM/EN-US/SQL/T-SQL/LANGUAGE-REFERENCE?VIEW=SQL-SERVER-VER15

(RDBMS) Relational Database Management System – History

https://en.wikipedia.org/wiki/Relational_database

Relational databases are based on the relational model of data, as proposed by E. F. Codd in 1970.

Edgar Frank "Ted" Codd (August 23, 1923 – April 18, 2003) was a British computer scientist and winner of the 1981 Turing Award.



Tools That Use T-SQL

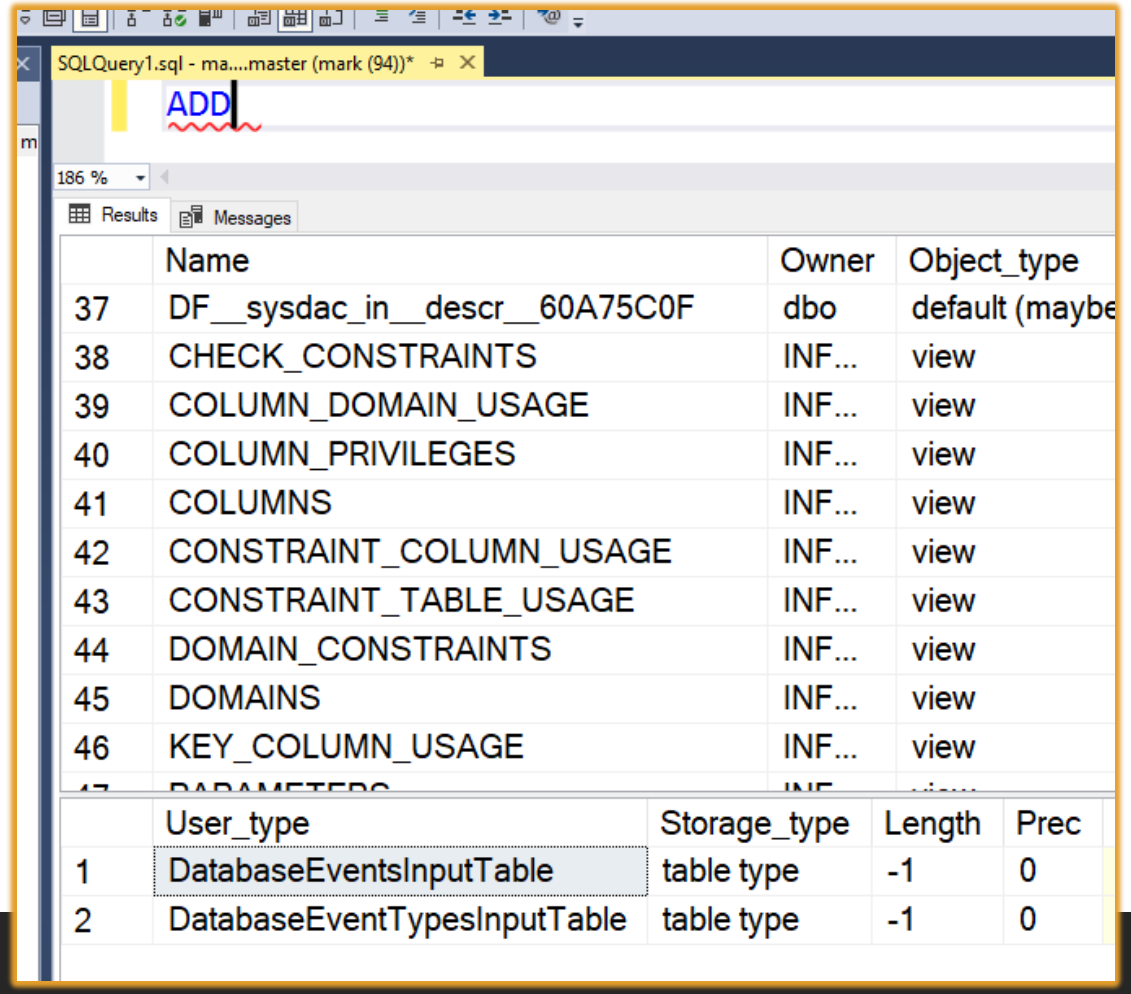
https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15#tools-that-use-t-sql

T-SQL is central to using Microsoft SQL products and services. All tools and that communicate with a SQL database send T-SQL commands. SQL works on top off T-SQL.

Some of the Microsoft tools that issue T-SQL commands are:

- SQL Server Management Studio (SSMS)
- SQL Server Data Tools (SSDT)
- Azure Data Studio

*You can type a T-SQL keyword in the SSMS Query Editor window and press F1 to get data about any T-SQL Keyword.



	Name	Owner	Object_type
37	DF__sysdac_in__descr__60A75C0F	dbo	default (maybe
38	CHECK_CONSTRAINTS	INF...	view
39	COLUMN_DOMAIN_USAGE	INF...	view
40	COLUMN_PRIVILEGES	INF...	view
41	COLUMNS	INF...	view
42	CONSTRAINT_COLUMN_USAGE	INF...	view
43	CONSTRAINT_TABLE_USAGE	INF...	view
44	DOMAIN_CONSTRAINTS	INF...	view
45	DOMAINS	INF...	view
46	KEY_COLUMN_USAGE	INF...	view

	User_type	Storage_type	Length	Prec
1	DatabaseEventsInputTable	table type	-1	0
2	DatabaseEventTypesInputTable	table type	-1	0

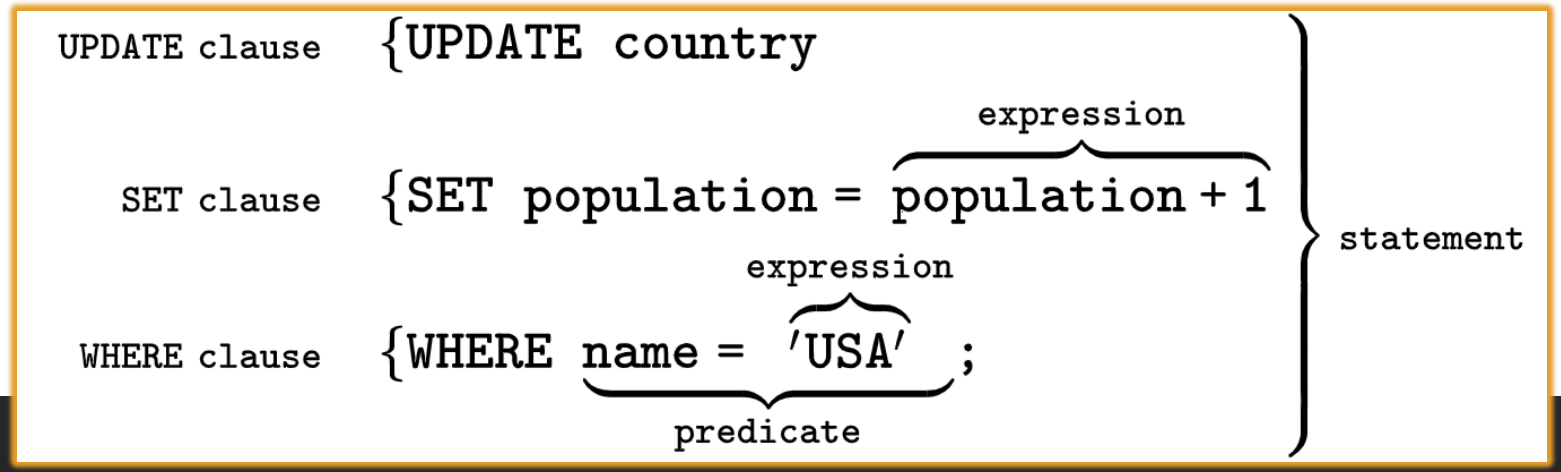
SQL (Structured Query Language)

https://en.wikipedia.org/wiki/SQL

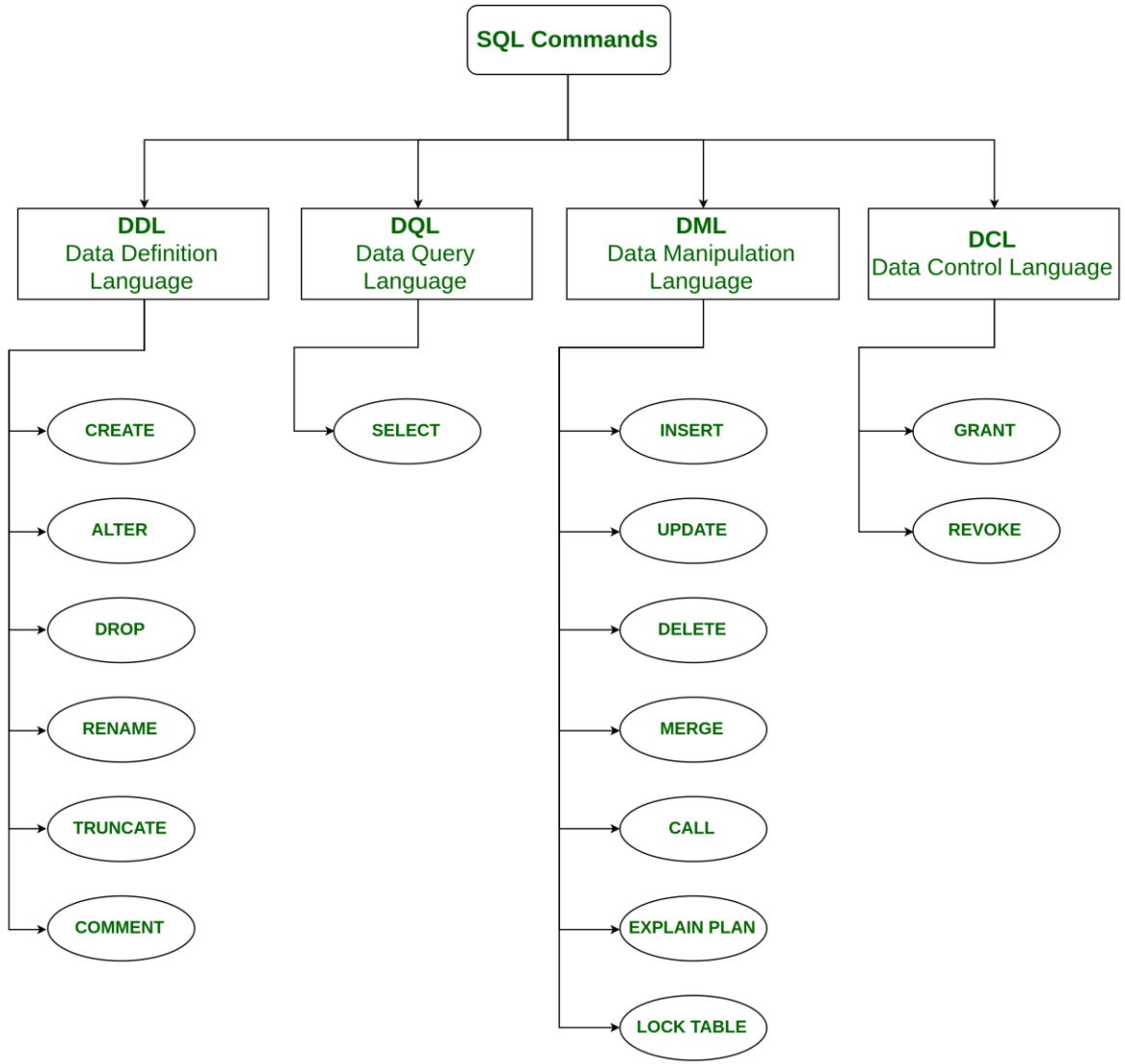
Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements which may be informally classed as sublanguages.
The most important sublanguages (for our usage) are:

- Data Definition Language (DDL)
- Data Manipulation Language (DML).

The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control.
SQL is a declarative language. We say what data we want, not how to get it. We cannot manage how SQL obtains the data.



Types of SQL Commands



Data Manipulation Language

https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver15#data-manipulation-language

---

Data Manipulation Language (DML) is vocabulary used to query a Database to retrieve and modify data from an SQL Database. Use ***Data Manipulation Language*** statements to ***INSERT***, ***UPDATE***, ***DELETE***, ***TRUNCATE***, and query the rows in the database.

INSERT – Adds one or more rows to a table or a view.

DELETE – Removes one or more rows from a table or view in SQL Server.

UPDATE – Changes existing data in a table.

TRUNCATE - Removes all rows from a table without deleting the table. Similar to the ***DELETE*** statement with no ***WHERE*** clause, but faster.

SQL - SELECT, FROM, WHERE

https://docs.microsoft.com/en-us/sql/t-sql/queries/select-transact-sql?view=sql-server-ver15#logical-processing-order-of-the-select-statement

---

The three most common keywords in SQL Queries are ***SELECT***, ***FROM***, and ***WHERE***.

- ***SELECT*** - Retrieves rows from the database
- ***FROM*** – Usually required on the ***SELECT*** statement. Specifies from which table(s) the results will come.
- ***WHERE*** - Specifies the search conditions for the rows returned by the query.

```
SELECT first_name, last_name FROM
student_details
WHERE id = 100;
```

SQL – UPDATE a table

https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql?view=sql-server-ver15
https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql?view=sql-server-ver15#a-using-a-simple-update-statement

Changes existing data in a table or view in SQL Server 2019. There are many variations to UPDATE. Check the Docs for more detail.

```
--UPDATE table_name
--SET column1 = value1, column2 = value2, ...
--WHERE condition;
UPDATE Employee
SET TITLE = 'Tech Lead'
Where FirstName = 'Nick';
```

```
USE AdventureWorks2012;
GO
UPDATE Person.Address
SET ModifiedDate = GETDATE();
```

```
USE AdventureWorks2012;
GO
UPDATE Sales.SalesPerson
SET Bonus = 6000, CommissionPct = .10, SalesQuota = NULL;
GO
```

UPDATE multiple rows simultaneously.

SQL – INSERT into a table

https://docs.microsoft.com/en-us/sql/t-sql/statements/insert-transact-sql?view=sql-server-ver15#BasicSyntax
https://docs.microsoft.com/en-us/sql/t-sql/statements/insert-transact-sql?view=sql-server-ver15#BasicSyntax

---

Adds one or more rows to a table or a view in SQL Server.

```
--INSERT INTO table_name (column1, column2, column3, ...)
--VALUES (value1, value2, value3, ...);
SELECT * FROM Employee;

INSERT INTO Employee (LastName, EmployeeId, FirstName, Title)
VALUES ('Escalona',(SELECT MAX(EmployeeId) FROM Employee) + 1, 'Nick', 'Master');

SELECT * FROM Employee
WHERE EmployeeId = 102;
```

SQL – DELETE from a table

https://docs.microsoft.com/en-us/sql/t-sql/statements/delete-transact-sql?view=sql-server-ver15
https://docs.microsoft.com/en-us/sql/t-sql/statements/delete-transact-sql?view=sql-server-ver15#BasicSyntax

---

Removes one or more rows from a table or view in SQL Server.

```
--DELETE FROM table_name WHERE condition;
UPDATE Employee
SET ReportsTo = 1
Where EmployeeId = 102;

DELETE FROM Employee
WHERE FirstName LIKE 'N_C%' AND ReportsTo = 102;
```

SQL - Queries

Queries are used to insert, retrieve, modify, and delete data in a SQL Database. SQL Queries have a specific syntax and structure.

```
SELECT TerritoryID, Name
FROM Sales.SalesTerritory
ORDER BY TerritoryID ;
```

```
SELECT e.BusinessEntityID, d.Name AS Department
FROM HumanResources.Employee AS e
CROSS JOIN HumanResources.Department AS d
ORDER BY e.BusinessEntityID, d.Name ;
```

```
SELECT DepartmentNumber,
    DepartmentName,
    ManagerID,
    ParentDepartmentNumber
FROM DEPARTMENT
FOR SYSTEM_TIME AS OF '2014-01-01'
WHERE ManagerID = 5;
```

Query Examples

What do these queries do?

```
SELECT EmployeeKey, LastName
FROM DimEmployee
WHERE LastName = 'Smith' ;
```

What do these queries do?

```
SELECT *
FROM Production.Product
ORDER BY Name ASC;
```

Query Examples – Aggregate Functions

What do these queries do?

```
SELECT SUM(Quantity)
FROM OrderDetails;
```

What do these queries do?

```
SELECT AVG(Price)
FROM Products;
```

Query Examples – Aggregate Functions

What do these queries do?

```
SELECT SUM(Quantity)
FROM OrderDetails;
```

What do these queries do?

```
SELECT COUNT(ProductID)
FROM Products;
```

Query Examples – Aggregate Functions

What do these queries do?

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

What do these queries do?

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

DML Activity

-- basic exercises in groups of 3 (Chinook database)

-- 1. List all customers (full names, customer ID, and country) who are not in the US

-- 2. List all customers from brazil

-- 3. List all sales agents

-- 4. Show a list of all countries in billing addresses on invoices.

-- 5. How many invoices were there in 2009, and what was the sales total for that year?

-- 6. How many line items were there for invoice #37?

-- 7. How many invoices per country?

-- 8. Show total sales per country, ordered by highest sales first.

DML Activity - Answers

```
SELECT * FROM Customer;
-- 1. list all customers (full names, customer ID, and country) who are not in the US
SELECT CustomerId, FirstName, LastName, Country
FROM Customer
WHERE Country != 'USA';
-- 2. list all customers from brazil
-- 3. list all sales agents
SELECT *
FROM Employee
WHERE Title LIKE '%Sales%Agen?%';
-- pattern matching with the LIKE operator
-- % - 0 to many of any character
-- [abc] - one of a, b, or c
-- _ - one of any character
-- 4. show a list of all countries in billing addresses on invoices.
SELECT DISTINCT BillingCountry
FROM Invoice;
ESC, BillingCountry;
```

```
-- SELECT DISTINCT means, after you get all the result rows,
--    remove duplicate rows (where ALL column values match)
-- 5. how many invoices were there in 2009, and what was the sales total for that
year?
--    (extra challenge: find the invoice count sales total for every year, using one
query)
SELECT SUM(Total) AS TotalAmount, COUNT(InvoiceId) AS InvoicesIn2009
FROM Invoice
--WHERE InvoiceDate BETWEEN '2009-01-01' AND '2010-01-01';
WHERE YEAR(InvoiceDate) = 2009;
SELECT YEAR(InvoiceDate) AS Year, SUM(Total) AS TotalAmount, COUNT(InvoiceId)
AS Invoices
FROM Invoice
GROUP BY YEAR(InvoiceDate);
-- 6. how many line items were there for invoice #37?
-- 7. how many invoices per country?
-- 8. show total sales per country, ordered by highest sales first.
SELECT BillingCountry, SUM(Total)
FROM Invoice
GROUP BY BillingCountry
ORDER BY SUM(Total) D
```