

SQL - Structured Query Language

Query language to create, maintain and retrieve data from relational database

DDL - Data Definition Language

Build and modify the structure of the database

- 1) Create 2) Drop 3) Alter 4) Rename 5) Truncate

Eg) Create Table Employee (E-Id Int, E-Name Varchar(20))

Rename Table Employee to Customer

Drop Table Employee

Truncate Table Employee

Alter Table Customer Add C-Id Int Primary Key

Alter Table Customer Drop E-Id

Alter Table Customer Modify E-Name Varchar(50)

Constraints - Specifies rules for data in a table

- 1) Product-id int Primary Key
- 2) Customer-id int Foreign Key References Customer (Customer-Id)
- 3) Product-price int Not Null
- 4) Product-name varchar(30) Unique
- 5) Product-rating int Check (Product-rating >= 2.5)
- 6) Product-category varchar(20) Default "Product"
- 7) Product-id int Auto-Increment

Difference between drop and truncate

Drop - Drops the whole structure in one go, all constraints are lost, rollback not possible

Truncate - Deletes all the rows, constraints are preserved, rollback is possible

Difference between char and varchar

char - Remaining space is padded with blanks

varchar - Remaining space is saved

Differences between Primary Key and Unique

- Primary Key - Only one primary key per table. No null values allowed
- Unique - There can be multiple unique columns. 1 null value is acceptable

SQL injection is a web hacking technique. It occurs when you ask user for input like username, and the user enters sql command which might unknowingly run on your database

DML - Data Manipulation Language

Insert, Update, Retrieve, Delete data from the table

- Eg)
- Insert into Employee values (1001, 'Mahesh')
 - Insert into Employee (Name, Eid) values ('Sachin', 1002)
 - Update Employee Set Eid = 1003 where name = 'Mahesh'
 - Delete from Employee where Name Is Null
 - Select * from Employee

Clauses

- 1) Select * from Employee where age > 50
 - 2) Select * from Employee having age = 50
- Where clause cannot be used with aggregates, having can be used
- Where clause is used prior to group by, having is used after group by. Where acts as a pre filter and having acts as post filter
- 3) Select * from Book where pages Between 290 and 300
 - 4) Select * from Author where country is ('In', 'Us', 'Ka')
 - 5) Select Distinct Departments from Staff

Alias - Alternate name to tables or columns for ease of access

Used when 2 tables are a part of the query

- 1) Select F. Name, D. Name from Faculty F, Department D where F. Code = D. Code

Select concat(Name, '(', left(Occupation, 1), ')') from Occupation Order by Name

- Order By Clause - Used for sorting (ascending / descending)
 Select * from Faculty Order By Faculty - Name
 (We can also use Order By 2 - Column Number starting from 1)
 Select * from Faculty Order By Faculty - Name Desc

- Aggregate Functions

1) Avg 2) Count 3) Max 4) Min 5) Sum

Other Functions

1) Round 2) Floor 3) ceil 4) Sqrt() 5) Square() 6) Abs()
 7) Mod() 8) Pow(4, 3) (= 64)

Eg) Select Avg (Saleprice / Quantity) from petsale where animal = "DOG"
 Select Round (Saleprice, 2) from petsale → Rounded to 2 decimal point
 Select * from Employees where salary < (Select Avg (Salary) from Employees)
 Select Count (City) - Count (Distinct City) from Employees

- Group By - Groups the retrieved data

Eg) Select Dept, Count (Student - id) as Total Students from Students Group By Dept
 Select Cust - id, Sum (Amount) as Total - Loan from Customer Group By Cust - Id Having Total - Loan > 5000

- Pattern Matching

1) Using Like Keyword

Eg) Select name from faculty where name like 'A%' - Starting with A
 ... where name like '%AS' - Ending with AS
 ... where name like '%Y%' - Contains Y
 ... where name not like '%A%' - Not containing A
 ... where name like '-A%' - 3rd character is A
 ... where name like '-a%e%h-' - Second letter a, contains e, 2nd last h
 Select length (animal) Ucase (animal) Lcase (animal) from Petsale
 Select left (city, 1) from station - 1 character from start
 Select right (city, 2) from station - 2 characters from end
 Replace (Salary, '0', '') - Removes 0's

2) Regular Expression

Select * from Users where L-name Regexp 'field' - Contains field
 ... where L-name '^field' - Starts with field
 ... where L-name 'field\$' - Ends with field
 ... where L-name regexp 'field|mai' - Contains field or mai
 ... where L-name regexp '[gim]e' - Contains g/i/m before e
 ... where L-name regexp '[o-h]e' - Contains o/h/x/d.../h before e

- Null, Null(), Isnull()

Select * from faculty where Subject is Null

Select * from faculty where Subject is Not Null

Select f-name, Null(Subject, 'Not Alloted')[Subject] from Faculty
 If any row has Subject as Null, Null will display it as Not Alloted

Select f-name, Coalesce(Sub1, Sub2, Sub3)[Major Subject] from Faculty
 If sub1 is not null, sub1 will be displayed, else if sub2 is not null
 sub2 will be displayed, else if sub3 is not null sub3 will be displayed
 as Major Subject

- DCL - Data Control Language

Allow or restrict user from accessing the data

1) Grant - Grant Access to user / set of users

2) Revoke - Revoke Access from a user / set of users

- TCL - Transaction Control Language

1) Commit - Changes made Permanent 2) Rollback - Changes are undone

- Date/Time Functions

Year(), Month(), Day(), Hour(), Minute(), Second()

Eg) Select Day(Saledate) from Petrol

Select (Saledate + 3 Days) from Petrol

Select Current-Date - Saledate from Petrol

- Joins - Used to Join Multiple Tables
- 1) Natural Join - The join condition is not explicitly specified. Join is done on the basis of common columns having same name and data type.

Select E.Employee-id, E.Employee-name, D.Dept-name from Employee E Natural Join Department D
- 2) Cross Join - Cartesian Product. Occurs due to invalid join condition.
- 3) Inner Join - Tables are joined according to search condition, only the matching values are displayed.

Select E.E-id, E.E-Name, D.Dept-Name from Employee E Inner Join Department D On E.Dept-Id = D.Dept-Id
- 4) Left Outer Join - Returns all rows for which join condition is true and in addition returns all rows from left table and displays subordinate values of table as null.
- 5) Right Outer Join - Same as left outer join, but the dominant table is the right table.
- 6) Full Outer Join - Both the tables are dominant.
- Clauses which can be used in nested queries - In, Any, All, Exists
- Set Operations - Union, Intersect, Except

Union removes duplicate rows, union all displays all rows simply
Intersect provides rows which are common to both select statements
Except provides rows which are not common to both select statements